FPT SOFTWARE

Project Document

Fresher Academy Management System

HCM24_CPL_NET_02	
Group members	34
Supervisor	Mr. Nguyễn Thế Vinh
Ext. Supervisor	N/A
Project code	

-Ho Chi Minh City, 22th January 2024-

Table of Contents

A. INTRODUCTION 32

1. Project Information 32

2. Introduction 32

3. Current Situation 33

4. Problem Definition 33
5. Proposed Solution 33
5.1. Feature functions 34
5.2. Advantages 34
5.3. Disadvantages 34
6. Functional Requirements 34
B. SOFTWARE PROJECT MANAGEMENT PLAN 37
1. Problem Definition 37
1.1. Name of this Project 37
1.2. Problem Abstract 37
1.3. Project Overview 37
1.3.1. Current Situation 37
1.3.2. The Proposed System 37
2. Project Organization 40
2.1. Software Process Model 40
2.2. Roles and Responsibilities 40
2.3. Tools and Techniques 46
3. Project Management Plan 46
3.1. Product Backlog 46
3.2. Deliverables 48
3.3. All Meeting Minus 49
4. Coding Convention 49
4.1. Naming Conventions 49
4.2. Layout Conventions 51
4.3. Commenting Conventions 51
C. SOFTWARE REQUIREMENT SPECIFICATION 52

1. User Requirement Specification 52
1.1. Guest Requirement 52
1.2. Super Admin Requirement 52
1.3. Class Admin Requirement 52
1.4. Trainer Requirement 52
1.5. External Interface Requirement 53
1.5.1. User interface 53
1.5.2. Hardware interface 53
1.5.3. Software interface 53
1.5.4. Communication protocol 53
2. System Requirement Specification 54
2.1. System Overview Use Case 54
2.2. List of Use Cases 55
2.2.1. User Management Over View Use Case 55
2.2.1.1. <Guest> Login (UC_01) 56
2.2.1.2. <Guest> Forgot Password (UC_02) 57
2.2.1.3. <Super Admin> Create User (UC_03) 59
2.2.1.4. <Super Admin> Edit User (UC_04) 62
2.2.1.5. <Super Admin> Change Role (UC_05) 64
2.2.1.6. <Super Admin> Change Status(UC_06) 66
2.2.1.7. <Super Admin> Soft User List (UC_07) 67
2.2.1.8. <Super Admin> Search User (UC_08) 68
2.2.1.9. <Super Admin> Filter User (UC_09) 70
2.2.1.10. <Super Admin> View User List (UC_10) 72
2.2.1.11. <Super Admin> View User's Permission List (UC_11) 74
2.2.1.12. <Super Admin> Update User's permission List (UC_12) 76
2.2.1.13. <Super Admin> Import User (UC_13) 78

2.2.1.14. <Super Admin> Import Role (UC_14) 80
2.2.1.15. <Super Admin, Class Admin, Trainer> View List of Class (UC_15) 83
2.2.1.16. <Super Admin> Import Role Permission (UC_16) 85
2.2.1.17. <Super Admin> Export User List (UC_17) 87
2.2.1.18. <Super Admin> Export User's Permission List (UC_18) 89
2.2.1.19. <User> User Calendar (UC_19) 91
2.2.2. Syllabus Management Over View Use Case 92
2.2.2.1. <Super admin, Class admin, Trainer> Add Syllabus (UC_20) 93
2.2.2.2. <Super admin, Class admin, Trainer> View Syllabus (UC_21) 95
2.2.2.3. <Super admin, Class admin> Import Syllabus (UC_22) 96
2.2.2.4. <Super admin, Class admin, Trainer> Export syllabus list (UC_23) 98
2.2.2.5. <Super admin, Class admin, Trainer> Edit Syllabus (UC_24) 99
2.2.2.6. <Super admin, Class admin, Trainer> Search Syllabus (UC_25) 101
2.2.2.7. <Super admin, Class admin, Trainer> Delete Syllabus (UC_26) 102
2.2.2.8. <Super admin, Class admin, Trainer> De-activate Syllabus (UC_27) 104
2.2.2.9. <Super admin, Class admin, Trainer> Duplicate Syllabus (UC_28) 105
2.2.2.10. <Super admin, Class admin, Trainer> Save as draft (UC_29) 107
2.2.2.11. <Super admin, Class admin, Trainer> Search by Date picker (UC_30) 108
2.2.2.12. <Super admin, Class admin, Trainer> Paginate Syllabus list (UC_31) . 111
2.2.2.13. <Super admin, Class admin, Trainer> Add unit training materials (UC_32) 112
2.2.2.14. <Super admin, Class admin, Trainer> Sort syllabus list (UC_33) 114
2.2.2.15. <Super admin, Class admin, Trainer> Show list per page (UC_34) 115
2.2.2.16. <Super admin, Class admin> Download syllabus template (UC_35) 116
2.2.3. Training Program Management Over View Use Case 118
2.2.3.1. <Super admin, Class admin, Trainer> Search training program (UC_36) 118
2.2.3.2. <Super admin, Class admin, Trainer> Filter training program (UC_36) 120
2.2.3.3. <Super admin, Class admin> Edit training program (UC_37) 122
2.2.3.4. <Super admin, Class admin> Duplicate training program (UC_38) 123

2.2.3.5. <Super admin, Class admin> Delete training program (UC_39) 125
2.2.3.6. <Super admin, Class admin> De-activate training program (UC_40) 127
2.2.3.7. <Super admin, Class admin>Import csv (UC_41) 129
2.2.3.8. <Super admin, Class admin, Trainer> Export training program (UC_42) 131
2.2.3.9. <Super admin, Class admin, Trainer>Sort training program list (UC_43) 132
2.2.3.10. <Super admin, Class admin, Trainer>Paginate training program list (UC_44) 134
2.2.3.11. <Super admin, Class admin>Download training program template (UC_45) 135
2.2.3.12. <Super admin, Class admin>Search syllabus training materials (UC_46) 137
2.2.1.12. <Super admin, Class admin>Change training materials file (UC_47)... 138
2.2.3.13. <Super admin, Class admin>Delete training program materials (UC_48) 140
2.2.3.14. <Super admin, Class admin>Create training program (UC_49) 142
2.2.3.15. <Super admin, Class admin, Trainer>Show list per page (UC_50) 144
2.2.4. Class Management Over View Use Case 145
2.2.3.1. <Super Admin, Class Admin, Trainer> View Class Detail (UC_51) 146
2.2.3.2. <Super Admin, Class Admin, Trainer> Search Class on List (UC_52) 147
2.2.3.3. <Super Admin, Class Admin> Create Class (UC_53) 149
2.2.3.4. <Super Admin, Class Admin> Delete syllabus (UC_54) 152
2.2.3.5. <Super Admin, Class Admin> Edit Class (UC_55) 154
2.2.3.6. <Super Admin, Class Admin, Trainer> Search Calendar by week (UC_56) 156
2.2.3.7. <Super Admin, Class Admin, Trainer> Filter Calendar by day (UC_57). 158
2.2.3.8. <Super Admin, Class Admin>Delete Class (UC_58) 159
2.2.3.9. <Super Admin, Class Admin, Trainer> Class Detail in Calendar (UC_59) 161
2.2.3.10. <Super Admin, Class Admin, Trainer> Filter Class List (UC_60) 162
2.2.3.11. <Super Admin, Class Admin> Duplicated Class (UC_60) 164
2.2.3.12. <Super Admin, Class Admin, Trainer> Search on Training Calendar (UC_61) 166
2.2.3.13. <Super Admin, Class Admin, Trainer> View Training Calendar (UC_62) 167

2.2.3.14. <Super Admin, Class Admin, Trainer> Sort Class (UC_63) 169
2.2.3.15. <Super Admin, Class Admin>Elasticsearch (UC_64) 171
2.2.3.16. <Super Admin, Class Admin, Trainer> Export Class List (UC_65) 173
2.2.5. Student Information Management Over View Use Case 175
2.2.5.1. <Super Admin, Class Admin>View list students in System (UC_66) 176
2.2.5.2. <Super Admin, Class Admin>View list students in class (UC_67) 177
2.2.5.3. <Super Admin, Class Admin>Edit the information and status of each student from student list in class (UC_68) 179
2.2.5.4. <Super Admin, Class Admin>View Each Student (UC_69) 181
2.2.5.5. <Super Admin, Class Admin>Import Data (UC_70) 182
2.2.5.6. <Super Admin, Class Admin>Edit Student from Student System List (UC_71) 185
2.2.5.7. <Super Admin, Class Admin>Edit the information and status of each student from student list in class (UC_72) 188
2.2.5.8. <Super Admin, Class Admin>Edit Student Status in batch_Class (UC_73) 190
2.2.5.9. <Super Admin, Class Admin>Edit Student Status in batch_System (UC_74) 192
2.2.5.10. <Super Admin, Class Admin>Export the information of each student from student list in system (UC_75) 195
2.2.5.11. <Super Admin, Class Admin>Export the information of each student from student list in class (UC_76) 196
2.2.5.12. <Super Admin, Class Admin>Delete of each student (UC_77) 198
2.2.5.13. <Super Admin, Class Admin>Add students in system(UC_78) 200
2.2.5.14. <Super Admin, Class Admin>Add students in class (UC_79) 202
2.2.5.15. <Super Admin, Class Admin> Add Class Information (UC_80) 205
2.2.6. Score Management Over View Use Case 206
2.2.6.1. <Super Admin, Class Admin, Trainer> Import list of trainee's score (UC_81) 206
2.2.6.2. <Super Admin, Class Admin, Trainer> Export list of trainee's score (UC_82) 209

2.2.6.3. <Super Admin, Class Admin, Trainer> Update the score list for each student (UC_83) 210
2.2.6.4. <Super Admin, Class Admin, Trainer> View the trainee's score, ranking and final assessment of all modules per class (UC_84) 212
2.2.6.5. <Super Admin, Class Admin, Trainer> View final studying result per trainee (UC_85) 213
2.2.6.6. <Super Admin, Class Admin, Trainer> Update status of certificate award (UC_86) 215
2.2.7. Reservation Management Over View Use Case 217
2.2.7.1. <Super Admin, Class Admin> Update list of students with status "Reservation" (UC_87) 218
2.2.7.2. <Super Admin, Class Admin> Terminate/ Remove status "Reservation" (UC_88) 220
2.2.7.3. <Super Admin, Class Admin> Find new class for student after "Reservation" (UC_89) 223
2.2.7.4. <Super Admin, Class Admin>Update status "Drop out" for student (UC_90) 227
2.2.7.5. <Super Admin, Class Admin>View list of students with status "Reservation" (UC_91) 229
2.2.7.6. <Super Admin, Class Admin>Export the information of each student from reserve list in system (UC_92) 230
2.2.7.7. <Super Admin, Class Admin>Import reserve list (UC_93) 232
2.2.8. Email Inform/Remind Over View Use Case 233
2.2.8.1. <Super Admin, Class Admin> Create email template list (UC_94) 234
2.2.8.2. <Super Admin, Class Admin> Update email template (UC_95) 237
2.2.8.3. <Super Admin, Class Admin> Search email template (UC_96) 238
2.2.8.4. <Super Admin, Class Admin> Send email to trainer (UC_97) 241
2.2.8.5. <Super Admin, Class Admin> Send email to reserved student (UC_98) 243
2.2.8.6. <Super Admin, Class Admin> Send email to student from class list (UC_99) 245

2.2.8.7. <Super Admin, Class Admin> Send email to student from student list (UC_100) 247
2.2.8.8. <Super Admin, Class Admin> Export email template (UC_101) 249
3. Software System Attribute 250
3.1. Usability 250
3.2. Reliability 250
3.3. Availability 250
3.4. Security 250
3.5. Maintainability 251
3.6. Portability 251
3.7. Performance 251
4. Conceptual Diagram 251
D. SOFTWARE DESIGN DESCRIPTION 253
1. System Architectural Design 253
1.1. Web application architecture description 253
2. Design Overview 254
3. Component Diagram 255
4. Detailed Description 256
4.1. Class diagram 256
4.2. Class diagram explanation 260
4.2.1. AssignmentSchemes 260
4.2.2. Assignment 261
4.2.3. Class 262
4.2.4. Email Send 264
4.2.5. Module 265
4.2.6. Quiz 266
4.2.7. Role 267

4.2.8. Score 268
4.2.9. Student 269
4.2.10. AttendeeType 271
4.2.11. ClassUser 271
4.2.12. DeliveryType 271
4.2.13. EmailSendStudent 272
4.2.14. EmailTemplate 272
4.2.15. fsu 273
4.2.16. Location 274
4.2.17. Major 274
4.2.18. OutputStandard 274
4.2.19. QuizStudent 275
4.2.20. ReservedClass 276
4.2.21. RolePermission 276
4.2.22. StudentClass 276
4.2.23. StudentModule 278
4.2.24. Syllabus 278
4.2.25. SyllabusDay 280
4.2.26. SyllabusUnit 281
4.2.27. TechnicalCode 282
4.2.28. TechnicalGroup 282
4.2.29. TrainingMaterial 283
4.2.30. TrainingProgram 284
4.2.31. TrainingProgramModule 285
4.2.32. UnitChapter 285
4.2.33. User 287

4.2.34. UnitChapter 288
4.3. Interaction diagram 289
4.3.1. User Management Diagram 289
4.3.1.1. <Guest> Login 289
4.3.1.2. <Guest> Forgot Password 290
4.3.1.3. <Super Admin> Edit User 290
4.3.1.4. <Super Admin>Create User 291
4.3.1.5. <Super Admin> Change User Role 291
4.3.1.6. <Super Admin> Change Status 292
4.3.1.7. <Super Admin> Soft User List 292
4.3.1.8. <Super Admin> Search User 292
4.3.1.9. <Super Admin> Filter User 293
4.3.1.10. <Super Admin> View User List 293
4.3.1.11. <Super Admin> View User's Permission List 294
4.3.1.12. <Super Admin> Update User's Permission List 294
4.3.1.13. <Super Admin> Import User List 295
4.3.1.14. <Super Admin> Import New Role 296
4.3.1.15. <Super Admin> Import Role Permission 296
4.3.1.16. <Super Admin> Export User List 297
4.3.1.17. <Super Admin> Export User's Permission List 297
4.3.1.18. <User> User Calendar 298
4.3.2. Syllabus Management Diagram 298
4.3.2.1. <Super Admin, Class Admin, Trainer> View list Syllabus 298
4.3.2.2. <Super Admin, Class Admin, Trainer> Search by Date picker 299
4.3.2.3. <Super Admin, Class Admin> Import syllabus 299
4.3.2.4. <Super Admin, Class Admin, Trainer> Create Syllabus 300
4.3.2.5. <Super Admin, Class Admin, Trainer> Add unit training materials 301
4.3.2.6. <Super Admin, Class Admin, Trainer> Duplicate Syllabus 301

4.3.2.7. <Super Admin, Class Admin, Trainer> De-activate Syllabus 302
4.3.2.8. <Super Admin, Class Admin, Trainer> Delete Syllabus 302
4.3.2.9. <Super Admin, Class Admin, Trainer> Edit Syllabus 303
4.3.2.10. <Super admin, Class admin, Trainer> Export syllabus list 303
4.3.3. Training Program Management Diagram 304
4.3.3.1. <Super Admin, Class Admin, Trainer> View list Training Program 304
4.3.3.2. <Super Admin, Class Admin, Trainer> Filter training program 305
4.3.3.3. <Super Admin, Class Admin> Edit training program 305
4.3.3.4. <Super Admin, Class Admin> Duplicate training program 306
4.3.3.5. <Super Admin, Class Admin> Delete training program 307
4.3.3.6. <Super Admin, Class Admin> De-activate training program 307
4.3.3.7. <Super Admin, Class Admin>Import csv 308
4.3.3.8. <Super admin, Class admin, Trainer> Export training program 308
4.3.3.9. <Super Admin, Class Admin> View syllabus training materials 309
4.3.3.10. <Super Admin, Class Admin> Create training program 309
4.3.3.11. <Super Admin, Class Admin> Upload new training program materials310
4.3.3.12. <Super Admin, Class Admin> Delete training program materials 310
4.3.3.13. <Super Admin, Class Admin> Edit training program materials 311
4.3.4. Class Management Diagram 311
4.3.4.1. <Super Admin, Class Admin, Trainer> View Training Calendar 311
4.3.4.2. <Super Admin, Class Admin, Trainer> View Class List 312
4.3.4.3. <Super Admin, Class Admin> Create Class 313
4.3.4.4. <Super Admin, Class Admin, Trainer> View Detail Class 313
4.3.4.5. <Super Admin, Class Admin, Trainer> Filter class list 314
4.3.4.6. <Super Admin, Class Admin> Delete Syllabus 314
4.3.4.7. <Super Admin, Class Admin, Trainer> Search Class on List 315
4.3.4.8. <Super Admin, Class Admin> Edit Class 315
4.3.4.9. <Super Admin, Class Admin, Trainer> Search Calendar by Week 316

4.3.4.10. <Super Admin, Class Admin, Trainer> Filter Calendar by Day 317
4.3.4.11. <Super Admin, Class Admin, Trainer> Sort Class 317
4.3.4.12. <Super Admin, Class Admin> Duplicated Class 317
4.3.4.13. <Super Admin, Class Admin, Trainer> Elasticsearch 318
4.3.4.14. <Super Admin, Class Admin, Trainer> Search Class 319
4.3.4.15. <Super Admin, Class Admin> Delete Class 319
4.3.4.16. <Super Admin, Class Admin, Trainer> Export Class List 320
4.3.5. Student Management Diagram 320
4.3.5.1. <Super Admin, Class Admin> List Student In System 320
4.3.5.2. <Super Admin, Class Admin> List Student In Class 321
4.3.5.3. <Super Admin, Class Admin> Edit Student Information Detail 321
4.3.5.4. <Super Admin, Class Admin> Edit Student In Batch 322
4.3.5.5. <Super Admin, Class Admin> Delete Student 322
4.3.5.6. <Super Admin, Class Admin> Add New Student 322
4.3.5.7. <Super Admin, Class Admin> Export Student List 323
4.3.5.8. <Super Admin, Class Admin> Import Student List 324
4.3.5.9. <Super Admin, Class Admin> Add New Class Information 324
4.3.6. Score Management Diagram 325
4.3.6.1. <Super Admin, Class Admin, Trainer> Import list of trainee's score 325
4.3.6.2. <Super Admin, Class Admin, Trainer> Update the score list for each student 325
4.3.6.3. <Super Admin, Class Admin, Trainer> View the trainee's score, ranking and final assessment of all modules per class 326
4.3.6.4. <Super Admin, Class Admin, Trainer> View final studying result per trainee 326
4.3.6.5. <Super Admin, Class Admin, Trainer> Update status of certificate award327
4.3.6.6. <Super Admin, Class Admin, Trainer> Export Student Score List 327
4.3.7. Reservation Management Diagram 328

4.3.7.1. <Super Admin, Class Admin> View list of students with status "Reservation" 328
4.3.7.2. <Super Admin, Class Admin> Update list of students with status "Reservation" 328
4.3.7.3. Super Admin, Class Admin> Terminate/Remove status "Reservation" 329
4.3.7.4. <Super Admin, Class Admin> Find new class for student after "Reservation" 329
4.3.7.5. <Super Admin, Class Admin>Update status "Drop out" for student 330
4.3.7.6. <Super Admin, Class Admin> Export Reserve List 330
4.3.7.7. <Super Admin, Class Admin> Import Reserve List 331
4.3.8. Email Inform/Remind Diagram 331
4.3.8.1. <Super Admin, Class Admin> Add Email Template 331
4.3.8.2. <Super Admin, Class Admin> Search Email Template 332
4.3.8.3. <Super Admin, Class Admin> Edit Email Template 332
4.3.8.4. <Super Admin, Class Admin> Send Email To Trainer 332
4.3.8.5. <Super Admin, Class Admin> Send Email To Reserve 333
4.3.8.6. <Super Admin, Class Admin> Send Email To Student 334
4.3.8.7. <Super admin, Class admin> Export Email Template list 334
5. User Interface Design 335
5.1. User Management Interface 335
5.1.1. Login Page 335
5.1.2. Forgot Password Page 336
5.1.3. User Management Page 339
5.1.4. User's Permission Page 344
5.1.5. User Calendar 347
5.2. Class Management Interface 348
5.2.1. Create Class 348
5.2.2. Selected Training Program and Show Syllabus 352

5.2.3. Edit Training Detail 354
5.2.4. Class List Page 358
5.2.5. Class Detail 364
5.2.6. Edit Class 369
5.3. Syllabus Management Interface 376
5.3.1. Import syllabus 376
5.3.2. View list syllabus 377
5.3.3. Syllabus menu 379
5.3.4. Syllabus manage 380
5.3.5. Create Syllabus-General 381
5.3.6. Create Syllabus-Outline 382
5.3.7. Create Syllabus-Edit Unit 384
5.3.8. Create Syllabus-Other 385
5.3.9. Create Syllabus-Alert popup 386
5.3.10. Training materials 387
5.4. Training Program Management Interface 388
5.4.1. Training Program List 388
5.4.2. Filter table 390
5.4.3. Program Management 391
5.4.4. Create Training Program 392
5.4.5. Add Syllabus 393
5.4.6. Training program manage 395
5.4.7. View program detail 396
5.4.8. View program 399
5.4.9. Training materials 400
5.4.10. Import Training Program 402

5.5. Student Management Interface 403
5.5.1. Display Student List By Class 403
5.5.2. Add Class Infor Popup 405
5.5.3. Edit Student 406
5.5.4. Import function 408
5.5.5. View system page 409
5.5.6. Student management pop-up 411
5.5.7. Confirm delete pop-up 411
5.5.8. Select column pop-up 412
5.5.9. Filter pop-up 413
5.5.10. Action pop-up 414
5.5.11. Add student page 415
5.5.12. Edit student from System 417
5.5.13. Edit Student Status in batch_Class. 420
5.6. Score Management Interface 421
5.6.1. Import Score 421
5.6.2. View the trainee's score, ranking and final assessment of all modules per class 422
5.6.3. View final studying result per trainee 425
5.6.4. Update the score list for each student 426
5.6.5. Update status of certificate award 427
5.7. Reservation Management Interface 429
5.7.1. Reserve List Page 429
5.7.2. Add New Reserved Student Page 431
5.7.3. ReClass Dialog 433
5.7.4. ReClass Recommendation 435
5.8. Email Inform/Remind Interface 437

5.8.1. Email Template List 437

5.8.2. Email template details 440

5.8.3. Send email to trainer 441

5.8.4. Send remind reserve email 443

5.8.5. Send email to student 445

E. SYSTEM TEST EVIDENCE 447

1. User Management Test 447

2. Syllabus Management Test 448

3. Training Program Management Test 448

4. Class Management Test 450

5. Student Management Test 451

6. Score Management Test 451

7. Reservation Management Test 452

8. Email Inform/Remind Test 452

List of Figures

Figure 1 - Scrum Development Model 40

Figure 2 - System overview use case 54

Figure 3 - Use Management Over View Use Case 55

Figure 4 - <Guest> Login Use Case 56

Figure 5 - <Guest> Forgot Password Use Case 57

Figure 6 - <Super Admin> Create User Use Case 59

Figure 7 - <Super Admin> Edit User Use Case 62

Figure 8 - <Super Admin> Change Role Use Case 64

Figure 9 - <Super Admin> Change Status Use Case 66

Figure 10 - <Super Admin> Soft User Use Case 67

Figure 11 - <Super Admin> Search User Use Case 68

Figure 12 - <Super Admin> Filter User Use Case 70

Figure 13 - <Super Admin> View User List Use Case 72

Figure 14 - <Super Admin> View User's Permission List Use Case 74

Figure 15 - <Super Admin> Update User's Permission List Use Case 76

Figure 16 - <Super Admin> Import User Use Case 78

Figure 17 - <Super Admin> Import Role Use Case 81

Figure 18 - <Super Admin, Class Admin, Trainer> View List of Class Use Case 83

Figure 19 - <Super Admin> Import Role Permission Use Case 85

Figure 20 - <Super Admin> Export User List Use Case 87

Figure 21 - <Super Admin> Export User's Permission List Use Case 89

Figure 22 - <User> User Calendar Use Case 91

Figure 23 - Syllabus Management Over View Use Case 93

Figure 24 - <Super admin, Class admin, Trainer> Add Syllabus 93

Figure 25 - <Super admin, Class admin, Trainer>View Syllabus 95

Figure 26 - <Super admin, Class admin> Import Syllabus 96

Figure 27 - <Super admin, Class admin, Trainer> Export syllabus list 98

Figure 28 - <Super admin, Class admin, Trainer> Edit Syllabus 99

Figure 29 - <Super admin, Class admin, Trainer> Search Syllabus 101

Figure 30 - <Super admin, Class admin, Trainer> Delete Syllabus 103

Figure 31 - <Super admin, Class admin, Trainer> De-activate Syllabus 104

Figure 32 - <Super admin, Class admin, Trainer> Duplicate Syllabus 106

Figure 33 - <Super admin, Class admin, Trainer> Save as draft 107

Figure 34 - <Super admin, Class admin, Trainer> Search by Date picker 109

Figure 35 - <Super admin, Class admin, Trainer> Paginate Syllabus list 111

Figure 36 - <Super admin, Class admin, Trainer> Add unit training materials 112

Figure 37 - <Super admin, Class admin, Trainer> Sort syllabus list 114

Figure 38 - <Super admin, Class admin, Trainer> Show list per page..... 115

Figure 39 - < Super admin, Class admin> Download syllabus template 116

Figure 40 - Training Program Management Over View Use Case 118

Figure 41 - <Super admin, Class admin, Trainer> Search training program 118

Figure 42 - <Super admin, Class admin, Trainer> Filter training program..... 120

Figure 43 - <Super admin, Class admin> Edit training program..... 122

Figure 44 - <Super admin, Class admin> Duplicate training program..... 124

Figure 45 - <Super admin, Class admin> Duplicate training program..... 125

Figure 46 - <Super admin, Class admin> De-activate training program 127

Figure 47 - <Super admin, Class admin>Import csv 129

Figure 48 - <Super admin, Class admin, Trainer> Export training program 131

Figure 49 - <Super admin, Class admin, Trainer>Sort training program list..... 132

Figure 50 - <Super admin, Class admin, Trainer>Paginate training program list..... 134

Figure 51 - <Super admin, Class admin>Download training program template 135

Figure 52 - <Super admin, Class admin>Search syllabus training materials 137

Figure 53 - <Super admin, Class admin>Change training materials file 138

Figure 54 - <Super admin, Class admin>Delete training program materials..... 140

Figure 55 - <Super admin, Class admin>Create training program 142

Figure 56 - <Super admin, Class admin, Trainer>Show list per page..... 144

Figure 57 - Class Management Over View Use Case..... 145

Figure 58 - <Super Admin, Class Admin, Trainer> View Class Detail Use Case..... 146

Figure 59 - <Super Admin, Class Admin, Trainer> Search Class on List Use Case..... 148

Figure 60 - <Super Admin, Class Admin> Create Class Use Case 149

Figure 61- <Super Admin, Class Admin> Delete syllabus Use Case 152

Figure 62 - <Super Admin, Class Admin> Edit Class Use Case 154

Figure 63 - <Super Admin, Class Admin, Trainer> Search Calendar by week Use Case 156

Figure 64 - <Super Admin, Class Admin, Trainer> Filter Calendar by day Use Case..... 158

Figure 65 - <Super Admin, Class Admin> Delete Class Use Case..... 160

Figure 66 - <Super Admin, Class Admin, Trainer> Class Detail in Calendar Use Case... 161

Figure 67- <Super Admin, Class Admin, Trainer> Filter Class List Use Case..... 163

Figure 68 - <Super Admin, Class Admin> Duplicated Class Use Case..... 164

Figure 69 - <Super Admin, Class Admin, Trainer> Search on Training Calendar Use Case166

Figure 70 - <Super Admin, Class Admin, Trainer> View Training Calendar Use Case... 167

Figure 71 - <Super Admin, Class Admin, Trainer> Sort Class Use Case..... 169

Figure 72 - <Super Admin, Class Admin> Elasticsearch Use Case 171

Figure 73 - <Super Admin, Class Admin, Trainer> Export Class List..... 173

Figure 74 - Student Information Management Over View Use Case 175

Figure 75 - <Super Admin, Class Admin > View list of students by system 176

Figure 76 - <Super Admin, Class Admin > View list of students by class..... 177

Figure 77 - <Super Admin, Class Admin> Edit the information and status of each student (access from student list in class detail) 179

Figure 78 - <Super Admin, Class Admin> View each student detail 181

Figure 79 - <Super Admin, Class Admin> Import Data 183

Figure 80 - <Super Admin, Class Admin > Edit Student Detail..... 185

Figure 81 - <Super Admin, Class Admin> Edit the information and status of each student from student list in class detail..... 188

Figure 82 - <Super Admin, Class Admin> Edit Student Status in batch_class 190

Figure 83 - <Super Admin, Class Admin> Edit Student Status in batch_system..... 193

Figure 84 - <Class Admin, Super Admin>Export Data Excel In System..... 195

Figure 85 - <Class Admin, Super Admin> Export Data Excel..... 196

Figure 86 - <Class Admin, Super Admin>Delete Student..... 198

Figure 87 - <Super Admin, Class Admin>Add Student System 200

Figure 88 - <Super Admin, Class Admin >Add Student In Class 203

Figure 89 - <Super Admin, Class Admin> Add Class Information..... 205

Figure 90 - Score Management Over View Use Case 206

Figure 91 - <Super Admin, Class Admin, Trainer> Import list of trainee's score 207

Figure 92 - <Super Admin, Class Admin, Trainer> Export list of trainee's score..... 209

Figure 93 - <Super Admin, Class Admin, Trainer>Update the score list for each student.210

Figure 94 - <Super Admin, Class Admin, Trainer>View the trainee's score, ranking and final assessment of all modules per class 212

Figure 95 - <Super Admin, Class Admin, Trainer> View the studying result record per trainee. 214

Figure 96 - <Super Admin, Class Admin, Trainer>Update status of certificate award .. 215

Figure 97 - Reservation Management Over View Use Case 218

Figure 98 - <Super Admin, Class Admin> Update list of students with status "Reservation" 218

Figure 99 - <Super Admin, Class Admin> Terminate/ Remove status "Reservation" ... 221

Figure 100 - <Super Admin, Class Admin> Find new class for student after "Reservation"223

Figure 101 - <Super Admin, Class Admin> Update status "Drop out" for student 227

Figure 102 - <Super Admin, Class Admin> View list of student with status "Reservation"229

Figure 103 - <Class Admin, Super Admin>Export Data Excel In System 231

Figure 104 - <Class Admin, Super Admin> Import Data Excel In System..... 232

Figure 105 - Email Inform/Remind Over View Use Case 234

Figure 106 - <Super Admin, Class Admin> Create new email template..... 235

Figure 107 - <Super Admin, Class Admin> Update email template 237

Figure 108 - <Super Admin, Class Admin> Search email template 239

Figure 109 - <Super Admin, Class Admin> Update email template 241

Figure 110 - <Super Admin, Class Admin> Send email to reserved student 243

Figure 111 - <Super Admin, Class Admin> Send email to student 245

Figure 112 - <Super Admin, Class Admin> Send email to student 247

Figure 113 - <Super Admin, Class Admin> Search email template 249

Figure 114 - Conceptual Diagram 251

Figure 115 - System architectural design 253

Figure 116 - Component Diagram 255

Figure 117 - Class diagram 256

Figure 118 - <Guest> Login 289

Figure 119 - <Guest> Forgot Password 290

Figure 120 - <Super Admin> Edit User 290

Figure 121 - <Super Admin> Create User 291

Figure 122 - <Super Admin> Change User Role 291

Figure 123 - <Super Admin> Change Status 292

Figure 124 - <Super Admin> Soft User List 292

Figure 125 - <Super Admin> Search User 293

Figure 126 - <Super Admin> Filter User 293

Figure 127 - <Super Admin> View User List 294

Figure 128 - <Super Admin> Change User Role 294

Figure 129 - <Super Admin> Update User's Permission List 295

Figure 130 - <Super Admin> Import User List 295

Figure 131 - <Super Admin> Import New Role 296

Figure 132 - <Super Admin> Import Role Permission 296

Figure 133 - <Super Admin> Export User List 297

Figure 134 - <Super Admin> Export User's Permission List 297

Figure 135 - <User> User Calendar 298

Figure 136 - <Super Admin, Class Admin, Trainer> View list Syllabus 298

Figure 137 - <Super Admin, Class Admin, Trainer> Search by Date picker 299

Figure 138 - <Super Admin, Class Admin> Import syllabus 300

Figure 139 - <Super Admin, Class Admin, Trainer> Create Syllabus 300

Figure 140 - <Super Admin, Class Admin, Trainer> Add unit training materials 301

Figure 141 - <Super Admin, Class Admin, Trainer> Duplicate Syllabus 301

Figure 142 - <Super Admin, Class Admin, Trainer> De-activate Syllabus 302

Figure 143 - <Super Admin, Class Admin, Trainer> Delete Syllabus 302

Figure 144 - <Super Admin, Class Admin, Trainer> Edit Syllabus 303

Figure 145 - <Super admin, Class admin, Trainer> Export syllabus list 303

Figure 146 - <Super Admin, Class Admin, Trainer> View list Training Program 304

Figure 147 - <Super Admin, Class Admin, Trainer> Filter training program 305

Figure 148 - <Super Admin, Class Admin> Edit training program 306

Figure 149 - <Super Admin, Class Admin> Duplicate training program 306

Figure 150 - <Super Admin, Class Admin> Delete training program 307

Figure 151 - <Super Admin, Class Admin> De-activate training program 307

Figure 152 - <Super Admin, Class Admin> Import csv 308

Figure 153 - <Super admin, Class admin, Trainer> Export training program 308

Figure 154 - <Super Admin, Class Admin> View syllabus training program 309

Figure 155 - <Super admin, Class admin> Create training program 309

Figure 156 - <Super admin, Class admin> Upload new training program materials 310

Figure 157 - <Super admin, Class admin> Delete training program materials 310

Figure 158 - <Super admin, Class admin> Edit training program materials 311

Figure 159 - <Super Admin, Class Admin, Trainer> View Training Calendar 312

Figure 160 - <Super Admin, Class Admin, Trainer> View Class List 312

Figure 161 - <Super Admin, Class Admin> Create Class 313

Figure 162 - <Super Admin, Class Admin, Trainer> View Detail List 314

Figure 163 - <Super Admin, Class Admin, Trainer> Filter Class List 314

Figure 164 - <Super Admin, Class Admin> Delete Syllabus 315

Figure 165 - <Super Admin, Class Admin, Trainer> Search Class on List 315

Figure 166 - <Super Admin, Class Admin> Edit Class 316

Figure 167 - <Super Admin, Class Admin, Trainer> Search Calendar by Week 316

Figure 168 - <Super Admin, Class Admin, Trainer> Filter Calendar by Day 317

Figure 169 - <Super Admin, Class Admin, Trainer> Sort Class 317

Figure 170 - <Super Admin, Class Admin> Duplicated Class 318

Figure 171 - <Super Admin, Class Admin, Trainer> Elasticsearch 319

Figure 172 - <Super Admin, Class Admin, Trainer> Search Class 319

Figure 173 - <Super Admin, Class Admin> Delete Class 320

Figure 174 - <Super Admin, Class Admin, Trainer> Export Class List 320

Figure 175 - <Super Admin, Class Admin> List Student In System 321

Figure 176 - <Super Admin, Class Admin> List Student In Class 321

Figure 177 - <Super Admin, Class Admin> Edit Student Information 321

Figure 178 - <Super Admin, Class Admin> Edit Student In Batch 322

Figure 179 - <Super Admin, Class Admin> Delete Student 322

Figure 180 - <Super Admin, Class Admin> Add New Student 323

Figure 181 - <Super Admin, Class Admin> Export Student List 323

Figure 182 - <Super Admin, Class Admin> Import Student List 324

Figure 183 - <Super Admin, Class Admin> Add New Class Information 324

Figure 184 - <Super Admin, Class Admin, Trainer> Import list of trainee's score 325

Figure 185 - <Super Admin, Class Admin, Trainer> Update the score list for each student325

Figure 186 - <Super Admin, Class Admin, Trainer> View the trainee's score, ranking and final assessment of all modules per class 326

Figure 187 - <Super Admin, Class Admin, Trainer> View final studying result per trainee326

Figure 188 - <Super Admin, Class Admin, Trainer> Update status of certificate award327

Figure 189 - <Super Admin, Class Admin, Trainer> Export export student score list... 327

Figure 190 - <Super Admin, Class Admin> View list of students with status “Reservation” 328

Figure 191 - <Super Admin, Class Admin> Update list of students with status “Reservation” 328

Figure 192 - <Super Admin, Class Admin> Terminate/Remove status “Reservation”.. 329

Figure 193 - <Super Admin, Class Admin> Find new class for student after “Reservation”329

Figure 194 - <Super Admin, Class Admin> Update status “Drop out” for student 330

Figure 195 - <Super Admin, Class Admin> Export Reserve List..... 330

Figure 196 - <Super Admin, Class Admin> Import Reserve List 331

Figure 197 - <Super Admin, Class Admin> Add Email Template 331

Figure 198 - <Super Admin, Class Admin> Search Email Template..... 332

Figure 199 - <Super Admin, Class Admin> Edit Email Template 332

Figure 200 - <Super Admin, Class Admin> Send Email To Trainer 333

Figure 201 - <Super Admin, Class Admin> Send Email To Reserved Student 333

Figure 203 - <Super admin, Class admin> Export Email Template List 334

Figure 204 - Login Page UI 1 335

Figure 205 – Forgot Password Page 1 336

Figure 206 – Forgot Password Page 2 337

Figure 207 - Forgot Password Page 3 337

Figure 208 - User Management Page UI 1 339

Figure 209 - Edit Menu - User Management Page UI 2 339

Figure 210 - Add User Pop Up - User Management Page UI 3 340

Figure 211 - Update User Pop Up - User Management Page UI 4 340

Figure 212 - Import User Pop Up - User Management Page UI 5 340

Figure 213 - Filter Menu - User Management Page UI 6 341

Figure 214 - User’s Permission Page UI 1 344

Figure 215 - Permission Menu - Page UI 2 345

Figure 216 - Import New Role Pop Up - User's Permission Page UI 3 345

Figure 217 - Import Update Permission Pop Up - User's Permission Page UI 4 345

Figure 218 - User Calendar 347

Figure 219 - Create Class 348

Figure 220 - Training program search field - Create Class 349

Figure 221 - Selected Training Program and Show Syllabus 352

Figure 222 - Edit Training Detail 354

Figure 223 - Pop-up add new syllabus - Edit Training Detail 355

Figure 224 - Select syllabus - Edit Training Detail 355

Figure 225 - Training program_syllabus - Edit Training Detail 355

Figure 226 - Class List Page 358

Figure 227 - Class Pop-up Management and Filter Class Pop-up - Class List Page 359

Figure 228 - Class Detail 364

Figure 229 - Edit class - Class Detail 364

Figure 230 - Trainer/Admin phone number and email - Class Detail 365

Figure 231 - Edit Class 370

Figure 232 - Time picker drop-down - Edit Class 370

Figure 233 - Pop-up month,year,date picker - Edit Class 371

Figure 234 - Month drop-down list - Edit Class 371

Figure 235 - Year drop-down list - Edit Class 372

Figure 236 - Import syllabus UI 376

Figure 237 - View list syllabus UI 377

Figure 238 - Syllabus menu UI 379

Figure 239 - Syllabus manage UI 380

Figure 240 - Create Syllabus-General UI 381

Figure 241 - Create Syllabus-Outline UI 382
Figure 242 - Create Syllabus-Edit Unit UI 384
Figure 243 - Create Syllabus-Other UI 385
Figure 244 - Create Syllabus-Alert popup UI 386
Figure 245 - Training materials UI 387
Figure 246 - Training Program List UI 388
Figure 247 - Filter table UI 390
Figure 248 - Program Manage UI 391
Figure 249 - Create Training Program UI 392
Figure 250 - Add syllabus UI 393
Figure 251 - Training program manage UI 395
Figure 252 - View program detail UI 396
Figure 253 - View program UI 399
Figure 254 - Training materials UI 400
Figure 255 - Import Training Program UI 402
Figure 256 - Student list by class 403
Figure 257 - Add Class Infor Popup 405
Figure 258 - Edit Student 406
Figure 259 - Import 1 function UI 408
Figure 260 - Import 2 function UI 408
Figure 261 - View system page 409
Figure 262 - Student management pop-up 411
Figure 263 - Confirm delete pop-up 411
Figure 264 - Select column pop-up 412
Figure 265 - Filter pop-up 413
Figure 266 - Action pop-up 414

Figure 267 - Add student page 415

Figure 268 - Edit student from System 417

Figure 269 - Edit Student Status in batch_Class 420

Figure 270 - Import Score 421

Figure 271 - View the trainee's score, ranking and final assessment of all modules per class 422

Figure 272 - View final studying result per trainee 425

Figure 273 - Update the score list for each student 426

Figure 274 - Update status of certificate award 427

Figure 275 - Reserve List Page 429

Figure 276 - Add New Reserved Student Page 431

Figure 277 - ReClass Dialog 433

Figure 278 - ReClass Recommendation 435

Figure 279 - Email Template List page UI 437

Figure 280 - Search Filter Menu UI 437

Figure 281 - Add Email Template UI 438

Figure 282 - Email Template Detail UI 440

Figure 283 - Trainer info UI 441

Figure 284 - Send email menu UI 442

Figure 285 - Send email preview UI 442

Figure 286 - Reserve action menu UI 443

Figure 287 - Send remind email UI 444

Figure 288 - Send remind email preview UI 444

Figure 289 - Send remind email UI 445

Figure 290 - Send remind email preview UI 446

Figure 291 - User Management Test 447

Figure 292 - Syllabus Management Test 448

Figure 293 - Training Program Management Test 1 448

Figure 294 - Training Program Management Test 2 449

Figure 295 - Training Program Management Test 3 449

Figure 296 - Training Program Management Test 4 450

Figure 297 - Class Management Test 450

Figure 298 - Student Management Test 451

Figure 299 - Score Management Test 451

Figure 300 - Reservation Management Test 452

Figure 301 - Email Inform/Remind Test 452

List of Tables

Table 1 - Hardware Requirements for Web development 39

Table 2 - Software Requirement 40

Table 3 - Roles and Responsibilities Details 46

Table 4 - Tools 46

Table 5 - Techniques 46

Table 6 - Product backlog 48

Table 7 - Deliverables 49

Table 8 - Description of Actor for Website 54

Table 9 - Conceptual Diagram Data Dictionar 252

Table 10 - Components Dictionary 256

Table 11 - Components Dictionary 260

Table 12 - <User Interface-Field> Login Page 335

Table 13 - <User Interface-Button/Hyperlinks> Login Page 336

Table 14 - <User Interface-Field> Forgot Password Page 338

Table 15 - <User Interface-Field> Forgot Password Page 338

Table 16 - <User Interface-Field> User Management Page 342

Table 17 - <User Interface-Button/Hyperlinks> User Management Page..... 344

Table 19 - <User Interface-Button/Hyperlinks> User's Permission Page 347

Table 20 - <User Calendar-Field> User Calendar 347

Table 21 - <User Calendar-Button/Hyperlinks> User Calendar..... 348

Table 22 - <User Interface-Field> Create Class 350

Table 23 - <User Interface-Button/Hyperlinks> Create Class..... 351

Table 24 - <User Interface-Field> Selected Training Program and Show Syllabus..... 353

Table 25 - <User Interface-Field> Edit Training Detail 357

Table 26 - <User Interface-Button/Hyperlinks> Edit Training Detail 358

Table 27 - <User Interface-Field> Class List Page 362

Table 28 - <User Interface-Button/Hyperlinks> Class List Page..... 363

Table 29 - <User Interface-Field> Class Detail 368

Table 30 - <User Interface-Button/Hyperlinks> Class Detail 369

Table 31 - <User Interface-Field> Edit Class 375

Table 32 - <User Interface-Button/Hyperlinks> Edit Class 376

Table 33 - <User Interface-Field> Import syllabus..... 377

Table 34 - <User Interface-Button/Hyperlinks> Import syllabus 377

Table 35 - <User Interface-Field> View list syllabus..... 378

Table 36 - <User Interface-Button/Hyperlinks> View list syllabus 379

Table 37 - <User Interface-Button/Hyperlinks> Syllabus menu..... 379

Table 38 - <User Interface-Button/Hyperlinks> Syllabus manage 380

Table 39 - <User Interface-Field> Create Syllabus-General 382

Table 40 - <User Interface-Field> Create Syllabus-Outline 383

Table 41 - <User Interface-Button/Hyperlinks> Create Syllabus-Outline 384

Table 42 - <User Interface-Button/Hyperlinks> Create Syllabus-Edit Unit 384

Table 43 - <User Interface-Field> Create Syllabus-Other 386

Table 44 - <User Interface-Button/Hyperlinks> Create Syllabus-Other 386

Table 45 - <User Interface-Button/Hyperlinks> Create Syllabus-Alert popup 387

Table 46 - <User Interface-Field> Training materials 387

Table 47 - <User Interface-Button/Hyperlinks> Training materials 388

Table 48 - <User Interface-Field> Training Program List 389

Table 49 - <User Interface-Button/Hyperlinks> Training Program List 390

Table 50 - <User Interface-Field> Filter table 391

Table 51 - <User Interface-Button/Hyperlinks> Filter table 391

Table 52 - <User Interface-Button/Hyperlinks> Program Management 392

Table 53 - <User Interface-Field> Create Training Program 392

Table 54 - <User Interface-Button/Hyperlinks> Create Training Program 392

Table 55 - <User Interface-Field> Add syllabus 394

Table 56 - <User Interface-Button/Hyperlinks> Add syllabus 395

Table 57 - <User Interface-Button/Hyperlinks> Training program manage 395

Table 58 - <User Interface-Field> View program detail 398

Table 59 - <User Interface-Button/Hyperlinks> View program detail 398

Table 60 - <User Interface-Field> View program 400

Table 61 - <User Interface-Field> Training materials 401

Table 62 - <User Interface-Button/Hyperlinks> Training materials 401

Table 63 - <User Interface-Field>Import Training Program 402

Table 64 - <User Interface-Button/Hyperlinks>Import Training Program 403

Table 65 - <User Interface-Field> Student list by class 403

Table 66 - <User Interface-Button/Hyperlinks> Student list by class 404

Table 67 - <User Interface-Field> Edit Student 405

Table 68 - <User Interface-Button/Hyperlinks> Edit Student 405

Table 69 - <User Interface-Field> Edit Student 407

Table 70 - <User Interface-Button/Hyperlinks> Edit Student 407

Table 71 - <User Interface-Button/Hyperlinks> Import function 409

Table 72 - <User Interface-Field>View system page 409

Table 73 - <User Interface-Button/Hyperlinks> View system page 410

Table 74 - <User Interface-Button/Hyperlinks> Student management pop-up 411

Table 75 - <User Interface-Button/Hyperlinks> Confirm delete pop-up 412

Table 76 - <User Interface-Button/Hyperlinks> Select column pop-up 413

Table 77 - <User Interface-Button/Hyperlinks> Filter pop-up 414

Table 78 - <User Interface-Button/Hyperlinks> Action pop-up 414

Table 79 - <User Interface-Field> Add student page 416

Table 80 - <User Interface-Button/Hyperlinks> Add student page 416

Table 81 - <User Interface-Field> Edit student from system 419

Table 82 - <User Interface-Button/Hyperlinks> Edit student from system 419

Table 84 - <User Interface-Button/Hyperlinks> Edit Student Status in batch_Class 421

Table 85 - <User Interface-Field> Import Score 422

Table 86 - <User Interface-Field> View the trainee's score, ranking and final assessment of all modules per class 424

Table 87 - <User Interface-Field> View final studying result per trainee 426

Table 88 - <User Interface-Field> Update the score list for each student 427

Table 89 - <User Interface-Field> Update status of certificate award 429

Table 90 - <User Interface-Field> Reserve List Page 430

Table 91 - <User Interface-Button/Hyperlinks> Reserve List Page 431

Table 92 - <User Interface-Field> Add New Reserved Student Page 433

Table 93 - <User Interface-Button/Hyperlinks> Add New Reserved Student Page 433

Table 94 - <User Interface-Field> ReClass Dialog 435

Table 95 - <User Interface-Button/Hyperlinks> ReClass Dialog 435

Table 96 - <User Interface-Field> ReClass Recommendation 436

Table 97 - <User Interface-Button/Hyperlinks> ReClass Recommendation..... 436

Table 98 - <User Interface-Field> Email Template List page 439

Table 99 - <User Interface-Button/Hyperlinks> Email Template List page..... 439

Table 100 - <User Interface-Field> Email Template Detail page 441

Table 101 - <User Interface-Button/Hyperlinks> Email Template Detail page..... 441

Table 102 - <User Interface-Field> Send email to trainer 443

Table 103 - <User Interface-Button/Hyperlinks> Send email to trainer..... 443

Table 104 - <User Interface-Field> Send email to reserve 445

Table 105 - <User Interface-Button/Hyperlinks> Send email to reserve..... 445

Table 106 - <User Interface-Field> Send email to student..... 446

Table 107 - <User Interface-Button/Hyperlinks> Send email to student..... 447

A. Introduction

1. Project Information

- Project name: Fresher Academy Management System
- Project Code:
- Product Type: Website
- Start Date: 22th January 2024
- End Date: 20th April 2024

2. Introduction

In this document, We will the Fresher Academy Management System (FAMS), a cutting-edge solution tailored to enhance the efficiency of managing fresher academies at FPT. With a surge in enrollment and program complexity, FAMS addresses administrative challenges through features such as user management, syllabus management, and training program and class management.

FAMS is the outcome of meticulous research and analysis, providing a user-friendly interface for administrators to efficiently handle user accounts, course

materials, and scheduling. Our four-month development process is outlined, showcasing FAMS's conceptualization, component designs, and workflows.

We believe FAMS will be instrumental in optimizing fresher academy operations, offering a reliable and efficient platform. As we pioneer this solution, our vision is to shape the future of fresher academies, fostering a conducive learning environment for administrators and participants alike.

3. Current Situation

FPT's fresher academies are grappling with a substantial surge in enrollment and a diversified program landscape, necessitating more efficient administrative solutions. The prevailing methods of managing user accounts, course materials, and class schedules are proving to be increasingly challenging, leading to potential inefficiencies and oversight risks.

In this context, the Fresher Academy Management System (FAMS) emerges as a tailored solution to address the evolving needs of fresher academies at FPT. FAMS is designed to streamline operations, alleviate manual workload, and enhance overall management effectiveness. The demand for a centralized system is evident, and FAMS aims to fill this gap by providing a holistic approach to user management, syllabus organization, and training program and class administration.

4. Problem Definition

Below are disadvantages of current situation:

- Limited Accessibility: Difficulty in accessing and managing user accounts, training program, syllabuses, classes and schedules.
- Lack of Transparency: Insufficient visibility into critical data, hindering informed decision-making for administrators.
- Manual Workload: Reliance on manual processes, contributing to inefficiencies and increased workload for administrators.
- Inadequate Monitoring: Limited tools for monitoring training program performance, hindering overall quality assessment.

5. Proposed Solution

- This application will help simplifies the handling of user accounts, training programs, and syllabus for smoother administration.
- This application will help maintains a complete history of training transactions, allowing for better oversight and analysis.
- This application will help offers user-friendly tools and guidance for handling various administrative duties, such as managing programs and schedules.
- This application will help introduces streamlined tools for effectively managing, reducing manual efforts.

5.1. Feature functions

- User Management: Admins efficiently create, control access, and manage user profiles for trainers, class admins, and general admins.
- Training Program Management: Plan, schedule, and coordinate training programs, defining details, assigning trainers, and managing syllabi.
- Syllabus Management: Easily create, update, and maintain syllabi for courses and training programs.
- Class Management: Streamline class creation, scheduling, and trainer assignments within training programs.
- Student Information Management: Manage student data, including creation, access control, and profile updates.
- Score Management: Import scores, view module scores, update awards, and identify eligible students.
- Reservation Management: Efficiently handle reserved students, from addition to reminders and class arrangements.
- Email Inform Remind: Utilize comprehensive email features, including templates, editing, and notifications.

5.2. Advantages

- Streamlining user accounts, training programs, and syllabi results in more efficient and smoother administration.
- Maintaining a complete history of training transactions allows for thorough oversight and informed analysis.

5.3. Disadvantages

- System requires user to have the Internet to communicate with it.
- User must login to use system.
- Users might face challenges if they are not familiar with the new administrative tools, potentially leading to delays in task execution.

6. Functional Requirements

Functional requirements are listed as below:

- User Management:
  - Login, logout, Authenticate.
  - Forgot password
  - View list of existed user / User listing screen.
  - Create user.
  - Update user.
  - Delete user.

    - Active/Deactive user.
    - Grant permission to user account.
    - View permission matrix.
    - Update permission to user.
    - Filter user.
    - Sort user.
    - Search user.
    - Add new feature.
    - Pagination user.
    - Import user list.
    - Import role list.
    - Import Role permission.
    - Export user list
- Training Program Management:
    - View training program list.
    - View detail training program.
    - Update training program.
    - Delete training program.
    - Import training program.
    - Export training program.
    - Duplicate training program.
    - Active/deactive training program.
    - Search training program.
    - Upload materials.
    - Delete materials.
    - Update materials.
    - Download materials.
- Syllabus Management:
    - Create/Save as draft syllabus.
    - Import syllabus.
    - Export syllabus.
    - Duplicate syllabus.
    - Edit syllabus.
    - Active/Deactive syllabus.
    - Delete syllabus.
    - Get List of syllabuses & paging.
    - Search, filter & sort.
    - View syllabus detail.
- Class Management:
    - Implement story list of class, pagination, filter, sort.
    - Implement story search class on list.
    - Implement story delete/active/deactive.

- Implement story create class.
- Implement story view class detail.
- Implement story class's budget management.
- Implement story class's detail management.
- Implement story view training calendar.
- Implement story search on training calendar.
- Implement story update training calendar.
- Change training program, add/remove syllabus, add activity, select trainer, add/remove trainees, location.
- Export class list.

- Student Information Management:
  - Implement Story List Students of System , List Students of Class.
  - Implement story view student detail.
  - Implement story import student list.
  - Implement story update student information, update status student (each, in batch).
  - Implement story disable student.
  - Export student list.
- Score Management:
  - Import score list of students.
  - Export score list of students.
  - View the trainee's score, ranking and final assessment of all modules per class.
  - Update the score list for each student.
  - Update status of certificate award.
- Reservation Management:
  - Implement story view list of students .
  - Implement story update list of students.
  - Implement story terminate/ remove status.
  - Implement story inform the student about successful reservation.
  - Implement story remind student after "Reservation" period.
  - Implement story find new class for student after "Reservation" .
  - Implement story update status "Drop out" for student.
  - Export reservation list
- Email inform/ remind:
  - View history of sending emails to each student in status "Reservation".
  - Send reservation email(s) to each student.
  - Send email(s) to trainers in each class.
  - View list of all available templates in system.
  - Create a new email template.
  - Export email list

B. Software Project Management Plan

1. Problem Definition

1.1. Name of this Project

- Official name: **Fresher Academy Management System.**
- Vietnamese name: **Hệ thống quản lý học viện Fresher.**
- Abbreviation: **FAMS.**

1.2. Problem Abstract

The present scenario in fresher academy management encounters notable challenges, demanding the formulation of a holistic solution. Chief among these challenges is the deficiency in streamlined administrative processes, affecting the handling of user accounts, training programs, and syllabus.

Moreover, administrative responsibilities, encompassing the management of programs and schedules, suffer from a dearth of user-friendly tools and guidance, potentially leading to operational inefficiencies. Compounding these challenges is the imperative need for effective tools to manage fresher academies and alleviate manual efforts.

In response to these complexities, the proposed solution, the Fresher Academy Management System (FAMS), emerges with the objective of furnishing a unified platform enriched with user-centric features. The primary goal is to elevate administrative processes, foster transparency, and adeptly navigate the ever-evolving landscape of fresher academies.

1.3. Project Overview

1.3.1. Current Situation

Below are the problems encountered in this project:

A. Data Overload: Coping with the extensive information related to user accounts, training programs, and syllabus poses a significant challenge. Efficiently collecting and organizing this large volume of data remains a priority.

B. Operational Hurdles: The project faces operational obstacles in terms of ensuring smooth and streamlined processes for user authentication, program coordination, and syllabus management. These challenges require immediate attention and resolution.

1.3.2. The Proposed System

The Fresher Academy Management System (FAMS) aims to:

- Simplify data management for user accounts, training programs, and syllabus.
- Centralize resource management for easier access.
- Ensure scalability and adaptability for future growth.

Fresher Academy Management System (FAMS) Portal:

This portal allows users to:

- For Guest
  - Registration
  - Login
  - Forgot password
- For Super Admin
  - Manage User
  - Manage list of training programs
  - Manage list of syllabus
  - Manage list of classes
  - Manage student information
  - Manage score
  - Manage reservation student
  - Email inform remind
- For Class Admin
  - Manage list of training programs
  - Manage list of syllabus
  - Manage list of classes
  - Manage student information
  - Manage student scores
  - Manage reservation student
  - Email inform remind
- For Trainer
  - Manage list of syallbus except for import syllabus
  - View list of training programs
  - View training program detail
  - Search training programs by all field
  - Filter and sort training programs
  - View list of classes
  - View class details
  - Search classes by all fields
  - Filter and sort classes
  - Manage student scores

Boundaries of the System

- No mobile applications

Future Plan

In future we are intend to:

- Mobile App Integration: Develop mobile apps for convenient access.
- Dashboard Development: Create a dashboard to display various metrics and information about our training programs, syllabus, trainers, classes, students,

and users. This dashboard will provide an overview of key data points and insights for effective management and decision-making.

- Real-Time Web Push Notifications: Implement real-time notifications to instantly alert users about new updates, class changes, and urgent deadlines, enhancing engagement and communication.
- User calendar: Added function to create personal events, including: Event name, description, start date, end date, start time, end time, days of the week, improved UI, and only shows Event name, shows popup details after clicking on event name.
- Allow student access to application.
- Check Progress Features: A personal dashboard displays student progress, including hours studied, assignments completed, and progress against schedule.
- Make a dashboard to show all scores of all students in real-time (Real-Time Data Visualization), customization Dashboard, and edit to show any score the admin wants to see.
- Additional support for Vietnamese language

Development Environment

Hardware requirements

Hardware	Minimum	Recommended
Operating system	Windows XP	Windows 10 or above
Internet Connection	Cable, Wireless	Cable, Wireless
Computer Processor	Intel Core i3 Intel Core i5 or above	Intel Core i3 Intel Core i5 or above
Computer Memory	4GB	4GB 8GB or more

Table 1 - Hardware Requirements for Web development

Software requirements

Software	Name/ Version	Description
Operating System	Windows XP or above	Operating system and platform for development
IDE	Visual Studio Code and Visual Studio	Used for implement website
Design Model tool	StarUML v2.5.1	Used for creating model and diagram
DBMS	SQL Server	Used for create & manage the database for system
Document storage	GitLab	Used for storing document
Store and manage source code	GitLab	Used to store all source code
Communication	zalo, facebook, email	Used to exchange information, online meeting, communication
Web browser	Chrome 51 or above	Testing browser

Table 2 - Software Requirement

2. Project Organization

2.1. Software Process Model

In this project, we selected Scrum, part of the Agile software development model for managing product development. We chose Scrum because:

- Flexible: there is no permanent about requirement, time, easy to change or update
- Higher quality and lower risk
- Communicate between team member always put in high priority
- Time-saving: develop phase is faster
- Bugs or problems of project are detecting faster, easy for maintain

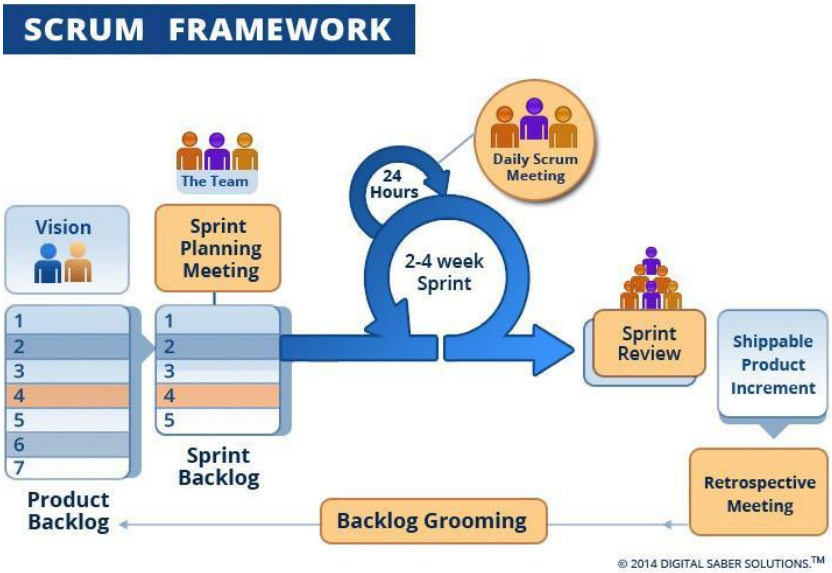


Figure 1 - Scrum Development Model

2.2. Roles and Responsibilities

No.	Full name	Role in Group	Responsibilities
1	Nguyễn Thế Vinh	Supervisor, Product owner	• Specify user requirements • Control the development process • Support in technique and business logic.

2	Lê Văn Đào	Team leader, Developer	• Clarify requirements • Design database. • Design GUI • Code • Write document and report
3	Lê Văn Nguyên	Team member, Developer	• Clarify requirements • Design database • Design GUI • Code • Write document and report
4	Mai Văn Quốc Tịnh	Team member, BA, Developer, Tester	• Clarify requirements • Design database. • Design GUI • Code • Write document and report
5	Lê Việt Hùng	Team member, Developer	• Clarify requirements • Design database. • Design GUI • Code • Write document and report
6	Nguyễn Công Thành	Team member, Developer	• Clarify requirements • Design database. • Design GUI • Code • Write document and report
7	Lê Văn Nguyên	Team member, Developer	• Clarify requirements • Design database. • Design GUI • Code • Write document and report

8	Lê Nguyễn Hoàng Phúc	Team Leader, BA, Developer	• Control the development process • Support in technique and business logic. • Create system architecture design • Design database. • Design GUI • Code • Write documents and report
9	Mai Tiến Hoàng	Team member, BA, Developer	• Manage process. • Create system architecture design. • Design database. • Design GUI • Code • Write documents and report
10	Ngô Quang Thắng	Team member, BA, Developer	• Manage process. • Create system architecture design. • Clarify requirements • Design database. • Design GUI • Code • Support in technique and business logic
11	Mã Ngọc Long Bảo	Team member, BA, Developer	• Manage process. • Create system architecture design. • Clarify requirements • Design database. • Design GUI • Code • Write documents and report
12	Nguyễn Công Danh	Team member, BA, Developer	• Manage process. • Create system architecture design. • Clarify requirements • Design database.

			• Design GUI • Code
13	Ngô Vỹ Khang	Team member, BA, Developer	• Manage process. • Create system architecture design. • Clarify requirements • Design database. • Design GUI • Code
14	Nguyễn Minh Hiền	Team member, Developer	• Code • Write document and report
15	Nguyễn Sơn Lam	Team member, Developer, Tester	• Code • Fix bug • Write document and report • Design GUI • Design database • Clarify requirements
16	Nguyễn Đỗ Chung Quý	Team member, Developer,	• Code • Write document and report • Design GUI
17	Nguyễn Mạnh Cường	Team member, Scrum master, Developer,	• Clarify requirements • Code • Write document and report
18	Nguyễn Lê Viên Minh	Team member, Developer	• Code • Write document and report
19	Nguyễn Hiệp Phú	Team member, Developer,	• Code • Write document and report

		Tester	• Design GUI • Design Database • Clarify requirements
20	Nguyễn Minh Phúc	Team member, Developer, Tester	• Code • Write document and report • Design Database • Clarify requirements
21	Nguyễn Hồ Tân Tiến	Team leader, Developer	• Clarify requirements • Design database. • Code
22	Bùi Châu Sao	Team member, Developer	• Clarify requirements • Design GUI • Code
23	Bùi Ngọc Minh Tâm	Team member, Developer	• Clarify requirements • Design GUI • Code
24	Nguyễn Hoàng Thái	Team member, Developer	• Clarify requirements • Code • Write document and report
25	Lê Quang Trọng	Team member, Developer	• Specify user requirements • Control the development process • Code • Support in technique and business logic.
26	Nguyễn Huỳnh Anh Tài	Team member, Tester	• Clarify requirements • Design test case • Test
27	Lê Quang Dũng	Team member, Developer	• Specify user requirements • Control the development process • Code • Support in technique and business

			logic. • Make Documentation
28	Nguyễn Đức Anh	Team member, Developer	• Specify user requirements • Control the development process • Code • Support in technique and business logic. • Make Documentation
29	Nguyễn Kỳ Anh Minh	Team member, Developer, Tester	• Code • Create test plan. • Test
30	Nguyễn Cao Thành Tài	Team member, Developer, Tester	• Design GUI • Code • Create test plan. • Test
31	Nguyễn Đức Linh	Team member, Developer, Tester	• Clarify requirements • Code • Create test plan. • Test
32	Nguyễn Minh Quang	Team member, Developer, Tester	• Code • Create test plan. • Test
33	Nguyễn Minh Nhật	Team member, Developer, Tester	• Code • Create test plan. • Test
34	Nguyễn Hoàng Phát	Team member, Developer, Tester	• Code • Create test plan. • Test

35	Lê Xuân Phước	Team member, Developer, Tester	• Code • Create test plan. • Test

Table 3 - Roles and Responsibilities Details

2.3. Tools and Techniques

Tools		
Developing Tool	Visual Studio	- ASP .Net Core
	Visual Studio Code	- React JS, Typescript
Managing Database	- SQL Server	
Source Control	- GitLab	
Modeling Tool	StarUML v2.8.1	
Web Browser	Chrome 51	

Table 4 - Tools

Techniques		
Developing tool	Front end	- React JS 18
	Back end	- .Net Core 8
Web servers		- .Net Core 8
Database Management System		- SQL Server

Table 5 - Techniques

3. Project Management Plan

3.1. Product Backlog

Story ID	Features	Task ID	Task description	Sprint
1	Create Product Backlog	1.1	Create Product backlog	1
2	Create Introduction	2.1	Create Introduction document	1

	document	2.2	Review Introduction document	1
3	Create mock up UI	3.1	Searching and Result viewing UI	1
		3.2	Login and register UI	1
		3.3	Import list UI	1
		3.4	List of Student UI	1
		3.5	User list UI	1
		3.6	Training program list UI	1
		3.7	Syllabus list UI	1
		3.8	Class list UI	1
		3.9	Score list UI	1
		3.10	Reservation UI	1
		3.11	Email inform UI	1
4	Create Software Project Management Plan	4.1	Problem definition	1
		4.2	Project organization	1
		4.3	Project management plan	1
		4.4	Coding convention	1
5	Create Software Requirements Specifications	5.1	User Requirement Specification	3
		5.2	External Interface Requirement	3
		5.3	Use case diagram	2
		5.4	Software System Attributes	3
		5.5	Database diagram	3
6	Create Software Design Description	6.1	Design Overview	2
		6.2	System Architectural Design	4

		6.3	Component Diagram	4
		6.4	Detailed Description of Components	4
		6.5	Sequence Diagram	3
		6.6	User Interface Design	2
		6.7	Database Design	1,2,3
		6.8	Coding	2, 3, 4, 5, 6
7	Create Software Test Documentation	7.1	Test Plan	5
		7.2	Test Cases	5
		7.3	Checklists	5
8	Software User's Manual	8.1	Installation Guide	6
		8.2	User's Guide	6

Table 6 - Product backlog

3.2. Deliverables

No	Deliverable	Delivery date	Delivery location	Note
1	Introduction Document, Task list	22/01/2024	FSA	Report No.1
2	Software Project Management Plan	26/01/2024	FSA	Report No.2
3	Software Requirements Specification	26/01/2024	FSA	Report No.3
4	Software Design Description	10/02/2024	FSA	Report No.4
5	Software Test Documentation Guide	8/03/2024	FSA	Report No.5

	Implementation (Coding)			
6	Software User's Manual	24/03/2024	FSA	Report No.6

Table 7 - Deliverables

* For each sprint, deliverables are potentially shippable products, which can be a part of documents or runnable software applications which is the implementation of some program features.
* Each sprint, we decide have 2 week.

3.3. All Meeting Minus

Please prefer to "Meeting Minutes" folder in the CD

4. Coding Convention

4.1. Naming Conventions

- **Frontend:**
  - **Components:**
    + Use PascalCase for naming components.
    + Example: MyComponent, Header, Footer.
  - **Props:**
    + Use camelCase for naming props passed to components.
    + Example: isLoggedIn, userName, onSubmit.
  - **State**:
    + Use camelCase for naming state variables.
    + Example: isLoading, errorMessage, userData.
  - **Events**:
    + Use camelCase for naming event handler functions.
    + Prefix event handler functions with handle.
    + Example: handleClick, handleChange, handleSubmit.
  - **Functions and Methods:**
    + Use camelCase for naming functions and methods.
    + Choose descriptive names that accurately represent their purpose.
    + Example: calculateTotal, fetchUserData, updateProfile.
  - **Variables:**
    + Use camelCase for naming variables.
    + Choose meaningful names that describe the data they hold.
    + Example: userList, selectedItem, errorCount.
  - **Constants:**
    + Use uppercase snake_case for naming constants.

    + Use descriptive names that indicate their purpose.
    + Example: MAX_LENGTH, API_URL, ERROR_MESSAGES.
  - **CSS Classes:**
    + Use kebab-case for naming CSS classes.
    + Choose semantic names that describe the purpose or styling of the element.
    + Example: button-primary, header-container, form-input.
  - **File Names:**
    + Use PascalCase for naming component files.
    + Separate words with dashes for non-component files.
    + Example: MyComponent.js, App.css, utils.js.
- **Backend:**
  - **Name-space**: Use lowercase letters for name-space names that reflect the structure and purpose of your application. Use title case (capitalizing the first letter of each word) for multi-word name-spaces (e.g., `myapp.api.controllers`).
  - **Class:** Use Pascal Case for class names. Class names should be nouns or noun phrases that describe the functionality of the class (e.g., `User Controller`, `Product Service`).
  - **Interface**: Use the prefix "I" and Pascal Case for interface names (e.g., `IUserService`, `IRepository`).
  - **Method**: Use PascalCase for method names. Method names should be verbs or verb phrases that describe the action performed (e.g., `GetUser`, `CreateProduct`).

- **Property**: Use PascalCase for property names. Property names should be nouns or noun phrases that describe the property (e.g., `FirstName`, `IsAdmin`).
- **Variable**: Use camelCase for variable names. Variable names should be descriptive and meaningful (e.g., `count`, `messageText`).
- **Constant**: Use uppercase letters and underscores for constant names. Constant names should be descriptive and meaningful (e.g., `MAX_RETRY_ATTEMPTS`, `API_KEY`).
- **Enum**: Use Pascal Case for enum names and Pascal Case or uppercase letters for enum values (e.g., `UserRole`, `Status.Active`, `Status.Inactive`).
- **Event**: Use Pascal Case for event names. Event names should be nouns or noun phrases that describe the event (e.g., `UserLoggedIn`, `ProductPurchased`).

4.2. Layout Conventions

- Use the default Code Editor settings (smart indenting, four-character indents, tabs saved as spaces).
- Write only one statement per line.
- Write only one declaration per line.
- If continuation lines are not indented automatically, indent them one tab stop (four spaces).
- Add at least one blank line between method definitions and property definitions.
- Use parentheses to make clauses in an expression apparent, as shown in the following code.

4.3. Commenting Conventions

- A block comment should be preceded by a blank line to set it apart from the rest of the code.
- Very short comments can appear on the same line as the code they describe, but should be shifted far enough to separate them from the statements.
- If a comment can't be written in a single line, it should follow the block comment format.
- The // comment delimiter can comment out a complete line or only a partial line. It shouldn't be used on consecutive multiple lines for text comments; however, it can be used in consecutive multiple lines for commenting out sections of code.

C. Software Requirement Specification

1. User Requirement Specification

1.1. Guest Requirement

Guest is a person who doesn't have access to the system. Guest can use some functions in the system. To use all functions, guest must regiter an account and login. Guest should have these following abilities:

- Registration
- Login

1.2. Super Admin Requirement

Super Admin is the person who has permission to manage website:

- Manage User
- Manage list of training programs
- Manage list of syllabus
- Manage list of classes
- Manage student information
- Manage score
- Manage reservation student
- Email inform remind

1.3. Class Admin Requirement

Class Admin is the person who completed resisting an account with Admin and logging into system can use these functions:

- Manage list of training programs
- Manage list of syllabus
- Manage list of classes
- Manage student information
- Manage student scores
- Manage reservation student
- Email inform remind

1.4. Trainer Requirement

Trainer is the person who completed resisting an account and logging into system can use these functions:

- Manage list of syallbi except for import syllabus
- View list of training programs

- View training program detail
- Search training programs by all field
- Filter and sort training programs
- View list of classes
- View class details
- Search classes by all fields
- Filter and sort classes
- Manage student scores

1.5. External Interface Requirement

1.5.1. User interface

- User interface must be friendly and easy to use.
- Website must be responsive, suitable with all devices such as: laptop, tablet or smartphone.
- All functions are arranged and displayed clearly.

1.5.2. Hardware interface

- The system will use the standard hardware and data communications resources of a standard computer.
- Desktop, laptop, smartphone or tablet have ability to connect to the Internet.

1.5.3. Software interface

- The website work with Chrome (v25 or above), Firefox (v46.1) and Safari (v4 or above).

1.5.4. Communication protocol

- Use HTTP protocol for communication between mobile application and web service.
- Use HTTP protocol for communication between web browser and web server.

2. System Requirement Specification

2.1. System Overview Use Case

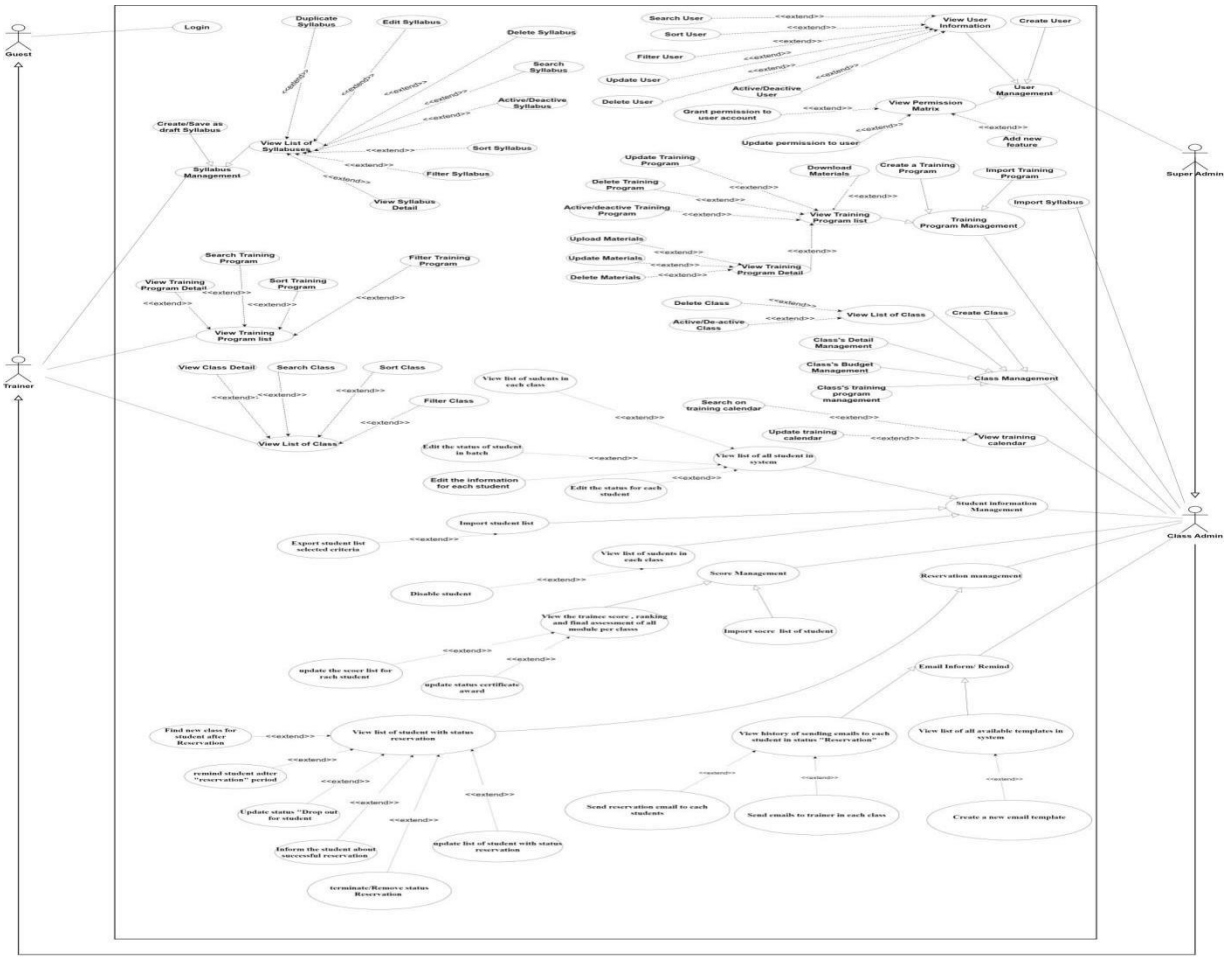


Figure 2 - System overview use case

No	Actor	Description	Note
1	Guest	Who does not have an account on system or not login into system yet	
2	Super Admin	Admin of system	
3	Class Admin	Admin responsible for managing classes and related activities within the system	
4	Trainer	Who already registered an account and login successfully into system	

Table 8 - Description of Actor for Website

2.2. List of Use Cases

2.2.1. User Management Over View Use Case

Use Case Diagram

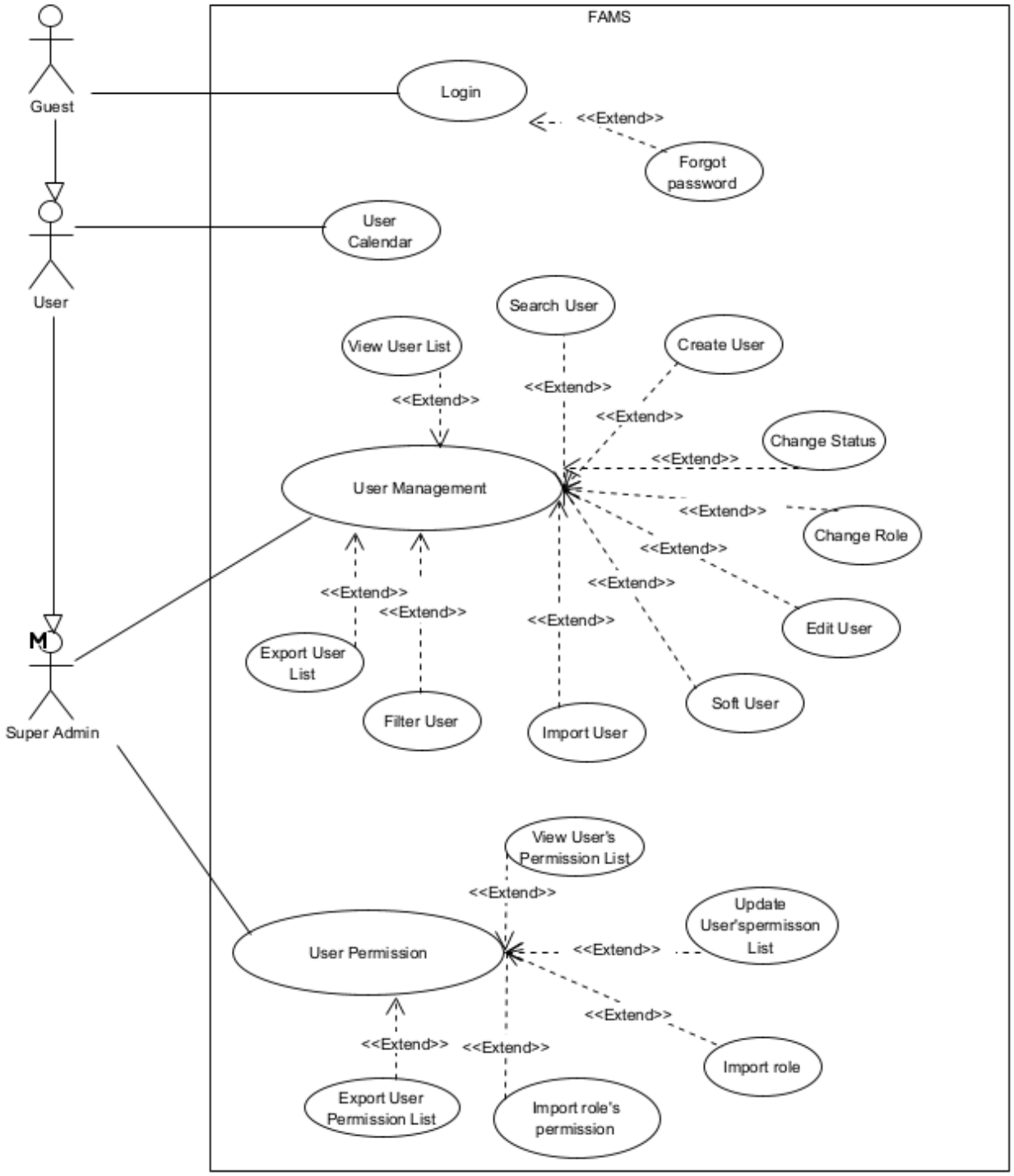


Figure 3 - Use Management Over View Use Case

2.2.1.1. <Guest> Login (UC_01)

Use Case Diagram

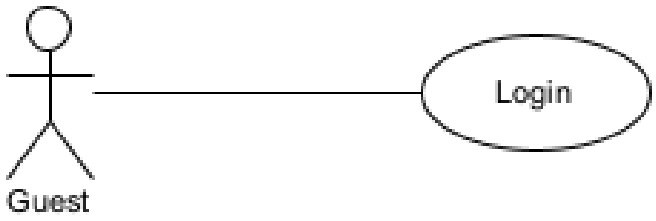


Figure 4 - <Guest> Login Use Case

Use Case Specification

USE CASE – UC_01			
Use Case No.	01	**Use Case Version**	2.0
Use Case Name	Login		
Author	Lê Văn Nguyên		
Date	18/03/2024	**Priority**	High

Actor:

- Guest

Summary:

- This use case allows the user to login to the website.

Goal:

- The user can access the website.

Triggers:

- The user want to use website.

Preconditions:

- The user must be have account.

Post Conditions:

- **Success:** The user can access the website.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response

1	User go to Login Page.	The system displays the screen with: + Username + Password
2	User enter username & password	System validates the information and shows a successful message.

Alternative: N/A

Exceptions:

No	Cause	System Response
1	"Username" is empty	Focus in "Phone" field. Show error message
2	"Password" is empty	Focus in "Password" field. Show error message

Relationships: N/A

Business Rules: N/A

2.2.1.2. <Guest> Forgot Password (UC_02)

Use Case Diagram

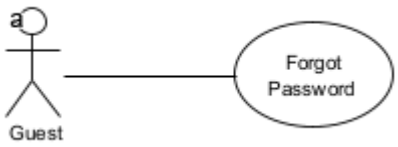


Figure 5 - <Guest> Forgot Password Use Case

Use Case Specification

USE CASE – UC_02			
Use Case No.	02	**Use Case Version**	2.0
Use Case Name	Forgot Password		
Author	Lê Văn Đào		
Date	16/4/2024	**Priority**	Normal

Actor:

- Guest

Summary:

- This use case allows a user to reset their password if they have forgotten it.

Goal:

- To enable users to securely reset their password and regain access to their account.

Triggers:

- User clicks on the "Forgot Password" link or button.

Preconditions:

- User is on the login page or the password reset page..

Post Conditions:

- **Success:** The user's password is successfully reset, and they can now log in with the new password.
- **Fail:** An error message is displayed, and the password reset process is interrupted.

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on the "Forgot Password" link or button.	The system displays the password reset page, prompting the user to enter their email address.
2	User enters their email address and clicks the "Reset Password" button.	The system generates a unique verification code and sends it to the user's email address.
3	User checks their email and enters the verification code on the password reset page.	The system verifies the code and displays the password reset form.
4	User enters a new password and confirms it, then clicks the "Reset Password" button.	The system updates the user's password in the system and logs the user in with the new password.

Alternative: N/A

Exceptions: If the system encounters any errors during the password reset process (e.g., email delivery failure, database connection issues, etc.), an error message is

displayed, and the password reset process is interrupted.

Relationships: This use case is related to the "Login" use case, as it allows users to regain access to their account.

Business Rules:

- The verification code must be a randomly generated 6-digit number.
- The verification code must expire after a certain period of time (e.g., 30 minutes) to prevent unauthorized access.
- The new password must meet the organization's password complexity requirements (e.g., minimum length, required character types).
- The system must ensure that the new password is not the same as the user's previous password.
- The system must log the password reset event and update the user's password change history.
- The system must provide clear and user-friendly error messages in case of any issues during the password reset process.
- The system must ensure that the password reset functionality is secure and resistant to various attacks (e.g., brute-force, phishing).

2.2.1.3. <Super Admin> Create User (UC_03)

Use Case Diagram

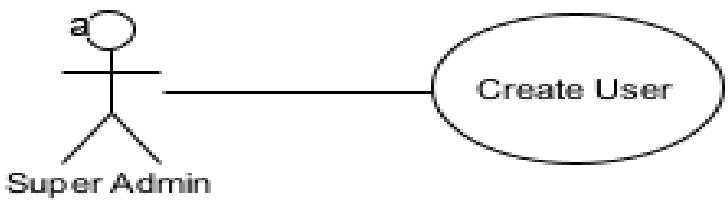


Figure 6 - <Super Admin> Create User Use Case

Use Case Specification

USE CASE – UC_03			
Use Case No.	03	**Use Case Version**	2.0
Use Case Name	Create user		
Author	Lê Văn Nguyên		
Date	18/03/2024	**Priority**	Normal
Actor: - Super Admin **Summary:**			

- This use case allows the admin to create a new account.

Goal:

- The account has been created.

Triggers:

- Admin click on "Add User" button.

Preconditions:

- Admin at "User Management" page.

Post Conditions:

- **Success:** User go to Login page with account and password have been registered.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click on "Create User" button.	The system request information in form: - Select Role: Choose 1 of 3 options: + Class Admin + Trainer + Super Admin - Full Name: Free text input but does not contain numbers or special characters. - Password: Have at least 12 characters. - Email: @gmail domain. - Address: Free text input. - Date of Birth: At least 18 years old. - Gender: Select options include: Male and Female.
2	Guest inputs information.	System validates the information and shows a successful message.

Alternative: N/A

Exceptions:

No	Cause	System Response
1	“Phone” is empty	Focus in “Phone” field. Show error message
2	“Full Name” is empty	Focus in “Full Name” field. Show error message
3	“Password” is empty	Focus in “Password” field. Show error message
4	“Password” length less than 12	Focus in “Password” field. Show error message
5	“Mail” is empty	Focus in “Email” field. Show error message
6	“Mail” is not “@gmail.com”	Focus in “Email field”. Show error message
7	“Address” is empty	Focus in “Address” field. Show error message
8	“Date of Birth” is empty	Focus “Date of Birth” field. Show error message
9	“Date of Birth” is under 18 age	Focus “Date of Birth” field. Show error message.
10	“Gender” is empty	Focus “Giới Tính” field. Show error message

Relationships: N/A

Business Rules:

- Email: Must be format “example@gmail.com”.
- Password length must be greater than 12.

2.2.1.4. <Super Admin> Edit User (UC_04)

Use Case Diagram

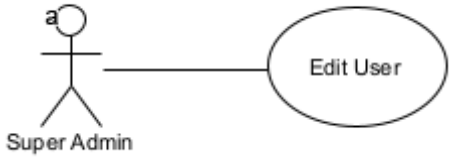


Figure 7 - <Super Admin> Edit User Use Case

Use Case Specification

USE CASE – UC_04			
Use Case No.	04	**Use Case Version**	2.0
Use Case Name	Edit User		
Author	Lê Văn Nguyên		
Date	18/03/2024	**Priority**	Normal

Actor:

- Super Admin

Summary:

- This use case allows the admin to edit an existing account.

Goal:

- The account will be updated with new information

Triggers:

- Admin click on the "..." button and select "Edit User".

Preconditions:

- Admin at "User Management" page.
- The account exists in the system.

Post Conditions:c

- **Success:** The information has been updated.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click on "..." button.	The system request information in form: + Edit user + Change role + De-activate user
2	Admin click on "Edit user".	The system displays the table include: - User Type: Choose 1 of 3 options: + Class Admin + Trainer + Super Admin - Full Name: Free text input but does not contain numbers or special characters.- Email Address: @gmail domain. - Date of Birth: At least 18 years old. - Gender: Select options include: Male and Female.
3	Admin click on "Save" button	The system validates and displays the result success or not.

Alternative: N/A

Exceptions:

No	Cause	System Response
1	"Phone" is empty	Focus in "Phone" field. Show error message
2	"Full Name" is empty	Focus in "Full Name" field. Show error message
3	"Mail" is empty	Focus in "Email" field. Show error message
4	"Mail" is not "@gmail.com"	Focus in "Email field". Show error message
5	"Date of Birth" is empty	Focus "Date of Birth" field. Show error message
6	"Date of Birth" is under 18 age	Focus "Date of Birth" field. Show error message.
7	"Gender" is empty	Focus "Giới Tính" field. Show error message

Relationships: N/A

Business Rules:

- Email: Must be format "example@gmail.com".

2.2.1.5. <Super Admin> Change Role (UC_05)

Use Case Diagram

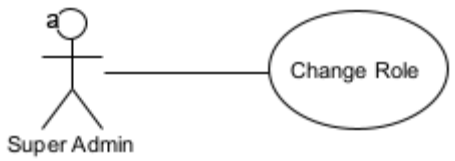


Figure 8 - <Super Admin> Change Role Use Case

Use Case Specification

USE CASE – UC_05			
Use Case No.	05	**Use Case Version**	2.0
Use Case Name	Change Role		
Author	Lê Văn Nguyên		
Date	18/03/2024	**Priority**	Normal

Actor:

- Super Admin

Summary:

- This use case allows the admin to change the role of an existing account.

Goal:

- The account will be updated with new information

Triggers:

- Admin click on the "..." button and select "Change Role".

Preconditions:

- Admin at "User Management" page.
- The account exists in the system.

Post Conditions:

- **Success:** The role of the user has been updated.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click on "..." button.	The system request information in form: + Edit user + Change role + De-activate user
2	Admin click on "Change Role".	The system displays the menu include: + Class Admin + Trainer + Super Admin
3	Admin select role	The system displays the result

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.1.6. <Super Admin> Change Status(UC_06)

Use Case Diagram

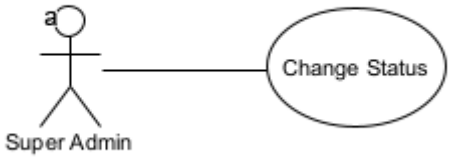


Figure 9 - <Super Admin> Change Status Use Case

Use Case Specification

USE CASE – UC_06			
Use Case No.	06	**Use Case Version**	2.0
Use Case Name	Change Status		
Author	Lê Văn Nguyên		
Date	18/03/2024	**Priority**	Normal

Actor:

- Super Admin

Summary:

- This use case allows the admin to change the status of an existing account.

Goal:

- The status of the user will be changed to active or inactive.

Triggers:

- Admin click on the "..." button and select "De-activate User" or "Activate User" base on their current status.

Preconditions:

- Admin at "User Management" page.
- The account exists in the system.

Post Conditions:

- **Success:** The status of the user has been updated.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click on "..." button.	The system request information in form: + Edit user + Change role + De-activate user
2	Admin click on "De-activate User".	The system displays the result and Switch button to "Activate User".

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.1.7. <Super Admin> Soft User List (UC_07)

Use Case Diagram

Figure 10 - <Super Admin> Soft User Use Case

Use Case Specification

USE CASE – UC_07			
Use Case No.	07	**Use Case Version**	2.0
Use Case Name	Sort User List		
Author	Lê Văn Nguyên		
Date	18/03/2024	**Priority**	Normal

Actor:

- Super Admin

Summary:

- This use case allows administrators to sort the list of users based on fields in descending or ascending order

Goal:

- The list will be sorted based on admin expectations.

Triggers:

- Admin clicks on the "filter" icon located next to the ID, Full Name, Email, Date of Birth, Gender and Type columns.

Preconditions:

- Admin at “User Management” page.
- There must exist data.

Post Conditions:

- **Success:** The status of the user has been updated.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin clicks on the "filter" icon located next to the ID, Full Name, Email, Date of Birth, Gender and Type columns.	The system displays a sorted list of users based on fields in descending or ascending order.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.1.8. <Super Admin> Search User (UC_08)

Use Case Diagram

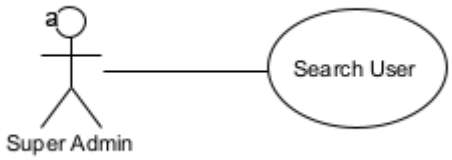


Figure 11 - <Super Admin> Search User Use Case

Use Case Specification

USE CASE – UC_08			
Use Case No.	08	**Use Case Version**	2.0
Use Case Name	Search User		
Author	Lê Văn Đào		
Date	19/03/2024	**Priority**	Normal

Actor:

- Super Admin

Summary:

- The use case allows the admin to search for users in the system using Elasticsearch.

Goal:

- .To provide the admin with an efficient way to search and retrieve user information from the system using Elasticsearch.

Triggers:

- The admin initiates the search for a user or group of users.

Preconditions:

- The admin is authenticated and authorised to access the system.
- Elasticsearch is set up and integrated with the system.

Post Conditions:

- **Success:** The admin successfully retrieves the user information from Elasticsearch.
- **Fail:** The system fails to retrieve user information from Elasticsearch.

Main Success Scenario:

Step	Actor Action	System Response

1	The admin initiates the search for a user or group of users by providing search criteria (e.g., name, email, or other user attributes).	The system queries Elasticsearch using the provided search criteria Elasticsearch returns the matching user records. The system displays the retrieved user information to the admin.

Alternative:

- If no matching user records are found, the system displays nothing.

Exceptions:

- If Elasticsearch is not available or experiences an error, the system displays an appropriate error message and logs the issue

Relationships:

- This use case is related to the "User Management" use case, as it provides a way to search and retrieve user information.

Business Rules:

- Only authorized super admins and admins can access and perform user searches.
- Elasticsearch should be optimized for efficient search and retrieval of user data.
- Search results should be displayed in a user-friendly and organized manner.

2.2.1.9. <Super Admin> Filter User (UC_09)

Use Case Diagram

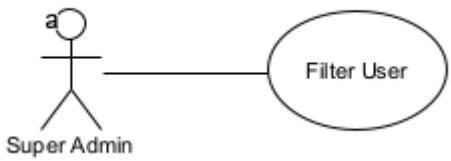


Figure 12 - <Super Admin> Filter User Use Case

Use Case Specification

USE CASE – UC_09			
Use Case No.	09	**Use Case Version**	2.0
Use Case Name	Filter User		
Author	Lê Văn Đào		
Date	19/03/2024	**Priority**	Normal

Actor:

- Super Admin

Summary:

- .This use case allows the admin to filter the list of users based on specific criteria such as Date of Birth (DOB), Gender, Type, and Status.

Goal:

- .To provide the admin with a flexible way to filter the user list according to their needs, enabling better user management and analysis.

Triggers:

- The admin initiates the filter operation on the user list.

Preconditions:

- The admin is authenticated and authorised to access the system.
- There is a list of users available in the system.

Post Conditions:

- **Success:** The system successfully filters the user list based on the specified criteria.
- **Fail:** The system fails to filter the user list.

Main Success Scenario:

Step	Actor Action	System Response
1	The admin selects the filter criteria, which can include Date of Birth (DOB), Gender (Male or Female), Type (Admin or User), and Status (Active or Inactive).	

		The system applies the specified filters to the user list.
		The system displays the filtered user list to the admin.

Alternative:

- If no filter criteria are specified, the system displays the complete user list without any filtering.

Exceptions:

- If the specified filter criteria are invalid or not supported, the system displays an appropriate error message.

Relationships:

- This use case is related to the "User Management" use case, as it provides a way to filter the user list for better management and analysis.

Business Rules:

- Only authorised admins can access and perform user list filtering.
- The system should provide a user-friendly interface for selecting filter criteria.
- The filtered user list should be displayed in a clear and organized manner.
- The system should handle large user datasets efficiently, ensuring reasonable response times for filter operations.

2.2.1.10. <Super Admin> View User List (UC_10)

Use Case Diagram

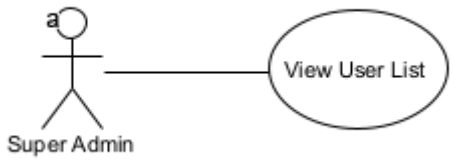


Figure 13 - <Super Admin> View User List Use Case

Use Case Specification

USE CASE – UC_10			
Use Case No.	10	**Use Case Version**	2.0
Use Case Name	View User List		
Author	Lê Văn Đào		
Date	19/03/2024	**Priority**	Normal

Actor:

- Super Admin

Summary:

- .This use case allows the admin to view the list of users in the system, displaying essential user information.

Goal:

- .To provide the admin with a comprehensive view of all registered users and their details.

Triggers:

- The admin requests to view the user list.

Preconditions:

- The admin is authenticated and authorized to access the system.
- There are registered users in the system.

Post Conditions:

- **Success:** The system successfully displays the user list with the required information.
- **Fail:** The system fails to retrieve or display the user list.

Main Success Scenario:

Step	Actor Action	System Response
1	The admin initiates the request to view the user list.	
2		The system retrieves the list of registered users from the database.

3		The system displays the user list, including the following information for each user: ID, Full Name, Email, Date of Birth, Gender (Male or Female), Type (Admin or Trainer).

Exceptions:

- If there is an error retrieving or displaying the user list, the system displays an appropriate error message and logs the issue.

Relationships:

- This use case is related to the "User Management" use case, as it provides a way to filter the user list for better management and analysis.

Business Rules:

- Only authorized admins can access and view the user list.
- The user list should be displayed in a user-friendly and organised manner, with appropriate sorting and filtering options.
- Sensitive user information, such as passwords, should not be displayed in the user list.
- The system should handle large user datasets efficiently, ensuring reasonable response times for displaying the user list.

2.2.1.11. <Super Admin> View User's Permission List (UC_11)

Use Case Diagram

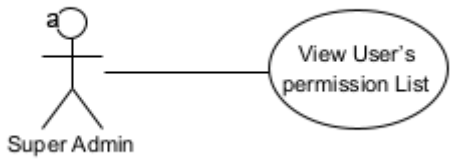


Figure 14 - <Super Admin> View User's Permission List Use Case

Use Case Specification

USE CASE – UC_11			
Use Case No.	11	**Use Case Version**	2.0
Use Case Name	View User's Permission List		
Author	Lê Văn Đào		
Date	19/03/2024	**Priority**	Normal

Actor:

- Super Admin

Summary:

- .This use case allows the admin to view the permissions granted to different user roles within the system.

Goal:

- .To provide the admin with a comprehensive view of the access and privileges assigned to user roles for various system components.

Triggers:

- The admin requests to view the user's permission list.

Preconditions:

- The admin is authenticated and authorized to access the system.
- User roles and permissions are defined and configured in the system.

Post Conditions:

- **Success:** The system successfully displays the user's permission list.
- **Fail:** The system fails to retrieve or display the user's permission list.

Main Success Scenario:

Step	Actor Action	System Response
1	The admin initiates the request to view the user's permission list.	
2		The system retrieves the configured user roles and their associated permissions.
3		The system displays the user's permission list, including the following information: Role Name, Syllabus, Training Program, Class, Learning Material, and User.
4		For each system component (Syllabus, Training Program, Class, Learning Material), the corresponding column displays a drop down indicating the level of access for that role: Access Denied, View, Modify, Create, or Full Access.

Alternative:

- If there are no user roles or permissions defined in the system, the system displays nothing.

Exceptions:

- If there is an error retrieving or displaying the user's permission list, the system displays an appropriate error message and logs the issue.

Relationships:

- This use case is related to the "User's Permission Management" use case, as it provides a way to filter the user list for better management and analysis.

Business Rules:

- Only authorised admins can access and view the user's permission list.
- The permission list should be displayed in a user-friendly and organised manner, with clear indicators for the level of access granted to each role.
- The system should support granular permission control, allowing for fine-grained access management across various system components.

Changes to user roles and permissions should be properly logged and audited for security and compliance purposes.

2.2.1.12. <Super Admin> Update User's permission List (UC_12)

Use Case Diagram

Figure 15 - <Super Admin> Update User's Permission List Use Case

Use Case Specification

USE CASE – UC_12			
Use Case No.	12	**Use Case Version**	2.0
Use Case Name	Update User's Permission List		
Author	Lê Văn Đào		
Date	19/03/2024	**Priority**	Normal
Actor: - Super Admin **Summary:**			

- .This use case allows the admin to update the permissions granted to different user roles within the system.

Goal:

- .To provide the admin with the ability to modify and configure the access and privileges assigned to user roles for various system components.

Triggers:

- The admin initiates the request to update the user's permission list.

Preconditions:

- The admin is authenticated and authorized to access the system.
- User roles and permissions are defined and configured in the system.

Post Conditions:

- **Success:** The system successfully updates the user's permission list based on the admin's changes.
- **Fail:** The system fails to update the user's permission list.

Main Success Scenario:

Step	Actor Action	System Response
1	The admin initiates the request to update the user's permission list.	
2		The system displays the current user's permission list, including the following information: Role Name, Syllabus, Training Program, Class, Learning Material, and User.
3	For each system component (Syllabus, Training Program, Class, Learning Material), the admin selects the desired level of access for a specific role from the dropdown: Access Denied, View, Modify, Create, or Full Access.	
4	The admin clicks on Update button.	

5		The system validates the changes and updates the user's permission list accordingly.
6		The system displays a success message confirming the update of the user's permission list.

Alternative:

- If the admin cancels the update operation, the system retains the existing user's permission list.

Exceptions:

- If there is an error updating the user's permission list, the system displays an appropriate error message and logs the issue.
- If the admin attempts to grant permissions that violate security policies or business rules, the system displays a warning and prevents the update.

Relationships:

- This use case is related to the "User's Permission Management" use case, as it provides a way to filter the user list for better management and analysis.

Business Rules:

- Only authorised admins can access and update the user's permission list.
- The system should enforce appropriate validation and security checks to ensure the integrity and consistency of the permission assignments.
- Changes to user roles and permissions should be properly logged and audited for security and compliance purposes.

The system should provide granular permission control, allowing for fine-grained access management across various system components.

2.2.1.13. <Super Admin> Import User (UC_13)

Use Case Diagram

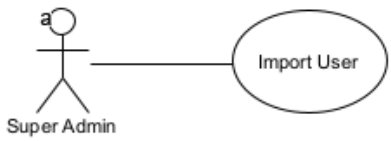


Figure 16 - <Super Admin> Import User Use Case

Use Case Specification

USE CASE – UC_13			
Use Case No.	13	**Use Case Version**	2.0
Use Case Name	Import User		
Author	Lê Văn Đào		
Date	19/03/2024	**Priority**	Normal

Actor:

- Super Admin

Summary:

- .This use case allows the admin to import user data into the system from a file with a predefined template.

Goal:

- .To provide the admin with a convenient way to bulk import user information from a file, streamlining the user management process.

Triggers:

- The admin initiates the import user process.

Preconditions:

- The admin is authenticated and authorised to access the system.
- The admin has access to a file containing user data in the required format (e.g., ImportUserTemplate.xlsx).

Post Conditions:

- **Success:** The system successfully imports the user data from the file.
- **Fail:** The system fails to import the user data from the file.

Main Success Scenario:

Step	Actor Action	System Response
1	The admin selects the option to import user data.	
2	The admin uploads or selects the file containing user data (e.g., ImportUserTemplate.xlsx).	

3		The system validates the file format and structure.
4		If the file is valid, the system imports the user data from the file into the system.
5		The system displays a success message indicating the number of users imported successfully.

Alternative:

- If the file format or structure is invalid, the system displays an error message and does not import the user data.

Exceptions:

- If there is an error during the import process (e.g., database connectivity issues, data conflicts), the system displays an appropriate error message and logs the issue.
- If the file contains duplicate or invalid user data, the system should handle these cases according to predefined rules (e.g., skip duplicates, import valid data only).

Relationships:

- This use case is related to the "User Management" use case, as it provides a way to filter the user list for better management and analysis.

Business Rules:

- Only authorised admins can access and perform the import user operation.
- The system should enforce validation rules to ensure the integrity and consistency of the imported user data.
- The import process should be optimised for efficient handling of large datasets.
- Appropriate error handling and logging mechanisms should be in place to track and resolve import issues.
- The system should provide a clear interface for selecting the import file and initiating the import process.

2.2.1.14. <Super Admin> Import Role (UC_14)

Use Case Diagram

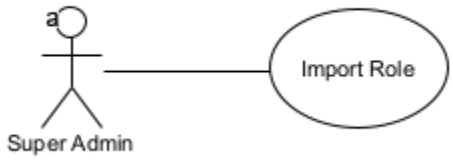


Figure 17 - <Super Admin> Import Role Use Case

Use Case Specification

USE CASE – UC_14			
Use Case No.	14	**Use Case Version**	2.0
Use Case Name	Import Role		
Author	Lê Văn Nguyên		
Date	19/03/2024	**Priority**	Normal

Actor:

- Super Admin

Summary:

- .This use case allows the admin to import a list of new roles into the system from a file with a predefined template.

Goal:

- .To provide the admin with a convenient way to bulk import user information from a file, streamlining the user management process.

Triggers:

- The admin initiates the import role..

Preconditions:

- The admin is authenticated and authorised to access the system.
- The admin has access to a file containing user data in the required format (e.g., ImportRole.xlsx).

Post Conditions:

- **Success:** The system successfully imports the role data from the file.
- **Fail:** The system fails to import the role data from the file.

Main Success Scenario:

Step	Actor Action	System Response

1	The admin selects the option to import role.	
2	The admin uploads or selects the file containing user data (e.g., ImportRoleTemplate.xlsx).	
3		The system validates the file format and structure.
4		If the file is valid, the system imports the user data from the file into the system.
5		The system displays a success message indicating the number of users imported successfully.

Alternative:

- If the file format or structure is invalid, the system displays an error message and does not import the role data.

Exceptions:

- If there is an error during the import process (e.g., database connectivity issues, data conflicts), the system displays an appropriate error message and logs the issue.
- If the file contains duplicate or invalid role data, the system should handle these cases according to predefined rules (e.g., skip duplicates, import valid data only).

Relationships: N/A

Business Rules:

- Only authorised admins can access and perform the import role.
- The system should enforce validation rules to ensure the integrity and consistency of the imported user data.
- The import process should be optimised for efficient handling of large datasets.
- Appropriate error handling and logging mechanisms should be in place to track and resolve import issues.

The system should provide a clear interface for selecting the import file and initiating the import process.

2.2.1.15. <Super Admin, Class Admin, Trainer> View List of Class (UC_15)

Use Case Diagram

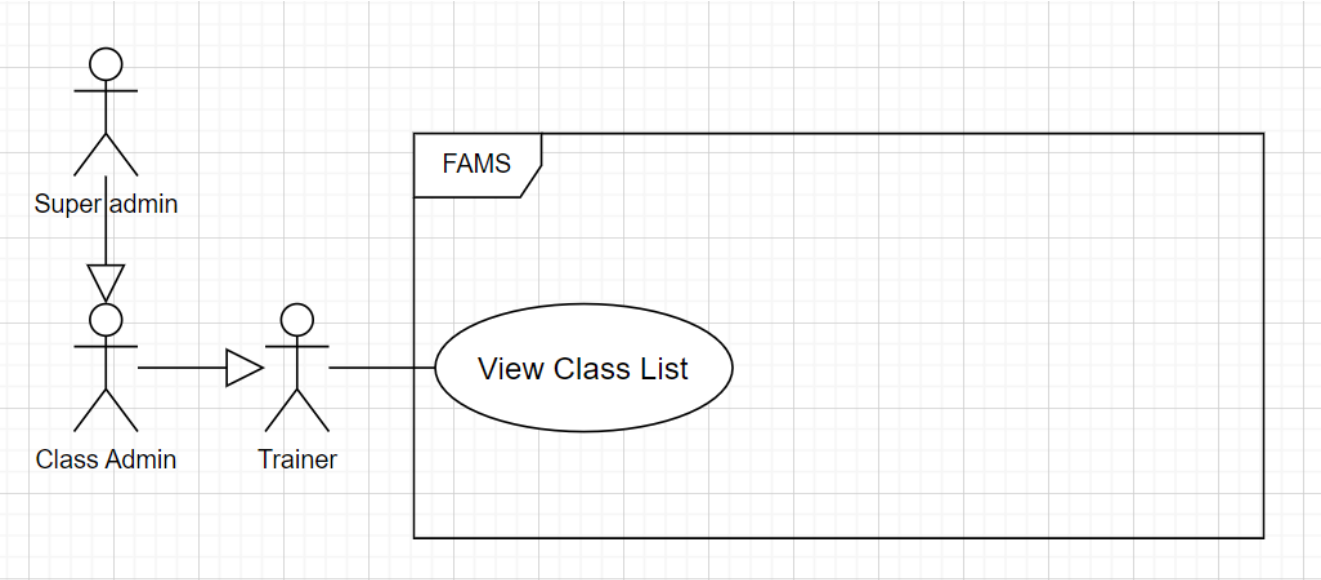


Figure 18 - <Super Admin, Class Admin, Trainer> View List of Class Use Case

Use Case Specification

USE CASE – UC_15			
Use Case No.	15	**Use Case Version**	2.0
Use Case Name	View list of class		
Author	Mai Văn Quốc Tịnh		
Date	18/03/2024	**Priority**	Normal
Actor: - Admin, Super Admin, Trainer. **Summary:** - This use case allows the admin, super admin, trainer to view a list of classes such as id, class, class code, create on, create by, duration, attendee, location, fsu. **Goal:** - To allow the admin, super admin, trainer to view detailed information about a specific class within the system. **Triggers:** - Admin, super admin, trainer click on the “Class” inside the bar and select “View Class”.			

Preconditions:

- Admin, super admin, trainer at "list of class" page.

Post Conditions:

- Success: Users go to the Login page with their account and password have been registered.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin, super admin, trainer selects a class from the list to view its details.	The system retrieves and displays comprehensive information about the selected class, including: +Id +Class name +Fsu +Location +Class code +Create on +Create by +Attendee
2	Admin, super admin, trainer reviews the displayed class details.	The system presents the class details accurately and clearly.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Class Filtering: Users should be able to filter classes based on various criteria such as id, class, class code, create on, create by, duration, attendee, location, fsu to quickly find the desired information.
- Sorting: Users should be able to sort the list of classes based on different parameters such as id, class, class code, create on, create by, duration, attendee, location, fsu for better organization.

2.2.1.16. <Super Admin> Import Role Permission (UC_16)

Use Case Diagram

Figure 19 - <Super Admin> Import Role Permission Use Case

Use Case Specification

USE CASE – UC_16			
Use Case No.	16	**Use Case Version**	2.0
Use Case Name	Import Role Permission		
Author	Lê Văn Nguyên		
Date	19/03/2024	**Priority**	Normal

Actor:

- Super Admin

Summary:

- This use case allows the admin to import a list of new role permissions into the system from a file with a predefined template.

Goal:

- To provide the admin with a convenient way to bulk import user information from a file, streamlining the user management process.

Triggers:

- The admin initiates the import role permission.

Preconditions:

- The admin is authenticated and authorised to access the system.
- The admin has access to a file containing user data in the required format (e.g., ImportRolePermission.xlsx).

Post Conditions:

- **Success:** The system successfully imports the role data from the file.
- **Fail:** The system fails to import the role data from the file.

Main Success Scenario:

Step	Actor Action	System Response
1	The admin selects the option to import role permissions.	
2	The admin uploads or selects the file containing user data (e.g., ImportRolePermissionTemplate.xlsx).	
3		The system validates the file format and structure.
4		If the file is valid, the system imports the user data from the file into the system.
5		The system displays a success message indicating the number of role permissions imported successfully.

Alternative:

- If the file format or structure is invalid, the system displays an error message and does not import the role data.

Exceptions:

- If there is an error during the import process (e.g., database connectivity issues, data conflicts), the system displays an appropriate error message and logs the issue.
- If the file contains duplicate or invalid role data, the system should handle these cases according to predefined rules (e.g., skip duplicates, import valid data only).

Relationships: N/A

Business Rules:

- Only authorized admins can access and perform the import role permission.
- The system should enforce validation rules to ensure the integrity and consistency of the imported user data.
- The import process should be optimized for efficient handling of large datasets.
- Appropriate error handling and logging mechanisms should be in place to track and resolve import issues.

The system should provide a clear interface for selecting the import file and initiating the import process.

2.2.1.17. <Super Admin> Export User List (UC_17)

Use Case Diagram

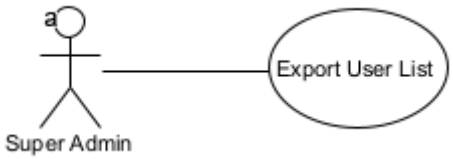


Figure 20 - <Super Admin> Export User List Use Case

Use Case Specification

USE CASE – UC_17			
Use Case No.	17	**Use Case Version**	2.0
Use Case Name	Export User List		
Author	Lê Văn Đào		
Date	15/4/2024	**Priority**	Normal

Actor:

- Super Admin, Admin

Summary:

- This use case allows the admin to export user list.

Goal:

- To generate an XLSX file containing the user list (excluding Super Admin users) that the admin can download.

Triggers:

- Admin click on “Export” button.

Preconditions:

- Admin at “User Management” page.

Post Conditions:

- **Success:** The XLSX file is generated and downloaded by the admin.
- **Fail:** An error message is displayed, and the export process is interrupted.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click on “Export” button.	The system initiates the export process.
2		The system generates an XLSX file containing the following columns: - Full Name - Email - Date of Birth (in the format "MM-dd-yyyy") - Address - Gender - Phone - Username - Status - Role (excluding "Super Admin")
3		The system prompts the admin to download the generated XLSX file.
4	Admin downloads the XLSX file.	The system completes the export process.

Alternative: N/A

Exceptions: If the system encounters any errors during the export process (e.g., database connection issues, invalid data, etc.), an error message is displayed, and the

export process is interrupted.

Relationships: N/A

Business Rules:

- The exported XLSX file must be named "ExportUserList.xlsx".
- The file must only include user records where the user's role is not "Super Admin".
- The date of birth must be in the format "MM-dd-yyyy".
- The exported data must accurately reflect the current state of the user records in the system, excluding the Super Admin users.
- The export process must be efficient and not cause significant performance degradation, even for large datasets.
- Appropriate error messages and feedback must be provided to the user in case of any issues during the export process.
- The export functionality must be secure and prevent unauthorized access to the user data.

The export button must be prominently displayed and easy to locate within the user management module.

2.2.1.18. <Super Admin> Export User's Permission List (UC_18)

Use Case Diagram

Figure 21 - <Super Admin> Export User's Permission List Use Case

Use Case Specification

USE CASE – UC_18			
Use Case No.	18	**Use Case Version**	2.0
Use Case Name	Export User's Permission List		
Author	Lê Văn Đào		
Date	15/4/2024	**Priority**	Normal
Actor: - Super Admin, Admin **Summary:** - This use case allows the admin to export the role permissions list.. **Goal:**			

- To generate an XLSX file containing the user list (excluding Super Admin users) that the admin can download.

Triggers:

- Admin click on “Export” button.

Preconditions:

- Admin is on the "Role Permissions" page.

Post Conditions:

- **Success:** The XLSX file is generated and downloaded by the admin.
- **Fail:** An error message is displayed, and the export process is interrupted.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click on “Export” button.	The system initiates the export process.
2		The system generates an XLSX file containing the following columns: - Role Name - Syllabus Permissions - Training Program Permissions - Class Permissions - Learning Material Permissions - User Management Permissions
3		The system prompts the admin to download the generated XLSX file.
4	Admin downloads the XLSX file.	The system completes the export process.

Alternative: N/A

Exceptions: If the system encounters any errors during the export process (e.g., database connection issues, invalid data, etc.), an error message is displayed, and the export process is interrupted.

Relationships: N/A

Business Rules:

- The exported XLSX file must be named "ExportRolePermissionsList.xlsx".
- The file must include all the role permissions data from the Role Permissions table.
- The role name and permission names must be displayed in a readable format (e.g., not just ID values).
- The exported data must accurately reflect the current state of the role permissions in the system.
- The export process must be efficient and not cause significant performance degradation, even for large datasets.
- Appropriate error messages and feedback must be provided to the user in

case of any issues during the export process.
- The export functionality must be secure and prevent unauthorized access to the role permissions data.

The export button must be prominently displayed and easy to locate within the role permissions management module.

2.2.1.19. <User> User Calendar (UC_19)

Use Case Diagram

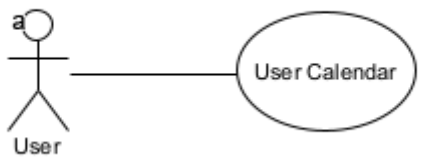


Figure 22 - <User> User Calendar Use Case

Use Case Specification

USE CASE – UC_19			
Use Case No.	19	**Use Case Version**	2.0
Use Case Name	User Calendar		
Author	Lê Văn Đào		
Date	15/4/2024	**Priority**	Normal

Actor:
- User

Summary:
- This use case allows the user to view their calendar of scheduled classes.

Goal:
- To provide the user with a calendar view of their upcoming and ongoing classes.

Triggers:
- User navigates to the "User Calendar" page.

Preconditions:
- User is logged in to the system.

Post Conditions:
- **Success:** The user's calendar is displayed with the scheduled classes
- **Fail:** An error message is displayed, and the calendar is not shown..

Main Success Scenario:

Step	Actor Action	System Response

1	User navigates to the "User Calendar" page.	The system retrieves the user's class information from the database.
2		The system generates a calendar view using the Big Calendar library, displaying the user's scheduled classes.
3	User interacts with the calendar (e.g., view events, navigate through different dates).	The system updates the calendar view accordingly.

Alternative: N/A

Exceptions:

- If the system encounters any errors while retrieving the user's class information (e.g., database connection issues, missing data), an error message is displayed, and the calendar is not shown.
- If the user does not have any scheduled classes, the calendar will be empty, and a message indicating that there are no scheduled classes will be displayed.

Relationships: N/A

Business Rules:

- The calendar view must display the following information for each scheduled class:
- Class name
- Class start and end dates
- Class start and end times
- FSU name
- Class location address
- The calendar must display the class status (e.g., "Ongoing", "Scheduled").
- The calendar must be interactive, allowing the user to navigate through different dates and view details of the scheduled classes.
- The calendar view must be responsive and optimized for both desktop and mobile devices.
- Appropriate error messages and feedback must be provided to the user in case of any issues during the calendar retrieval process.
- The calendar functionality must be secure and prevent unauthorized access to the user's class information.

The "User Calendar" page must be easily accessible and prominently displayed within the user management module.

2.2.2. Syllabus Management Over View Use Case

Use Case Diagram

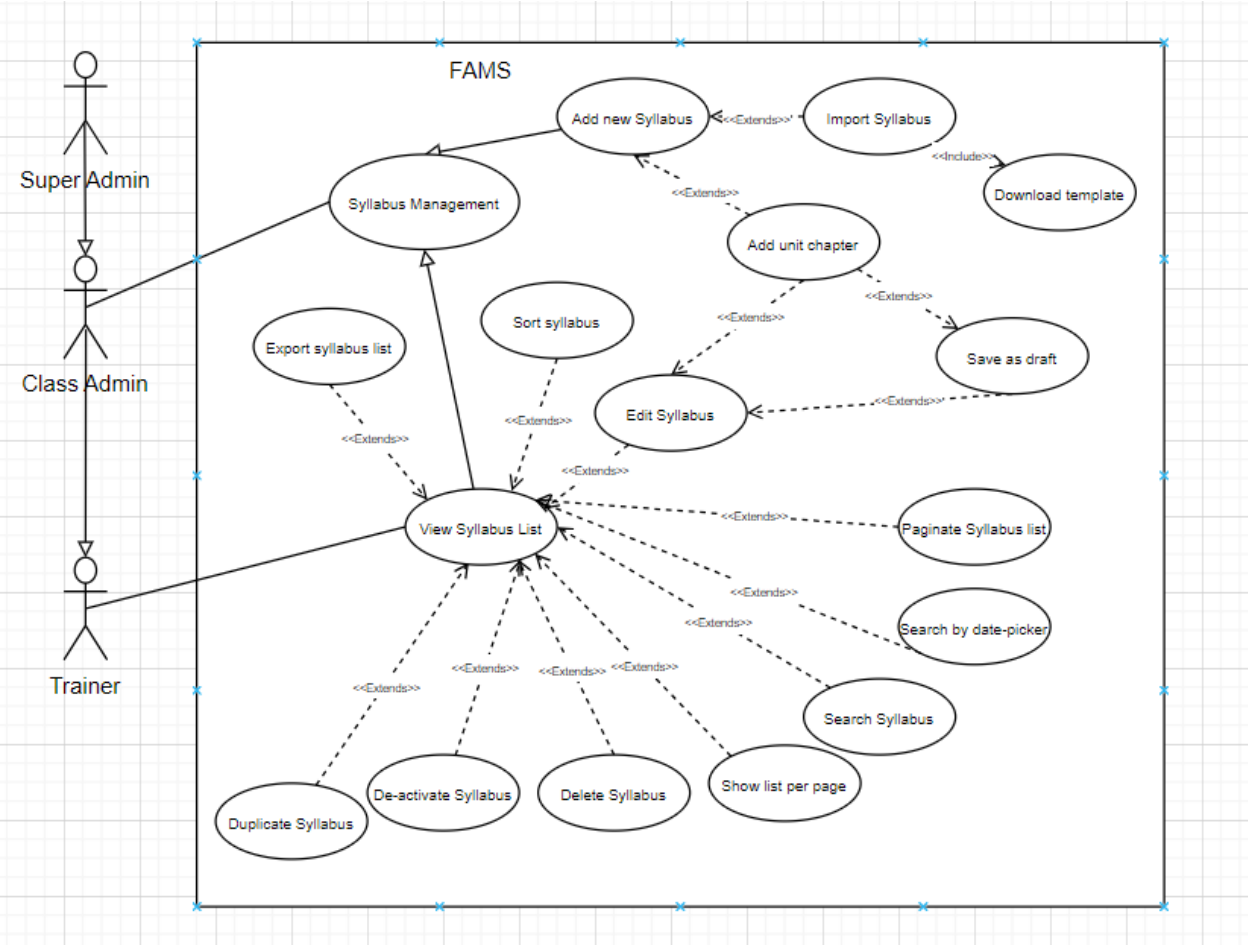


Figure 23 - Syllabus Management Over View Use Case

2.2.2.1. <Super admin, Class admin, Trainer> Add Syllabus (UC_20)

Use Case Diagram

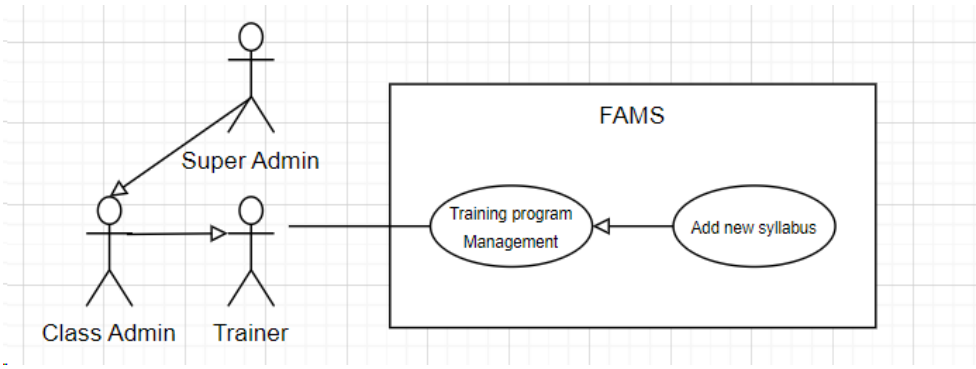


Figure 24 - <Super admin, Class admin, Trainer> Add Syllabus

Use Case Specification

USE CASE – UC_20			
Use Case No.	**20**	**Use Case Version**	**1.0**
Use Case Name	**Add Syllabus**		

Author	Mã Ngọc Long Bảo		
Date	19/03/2024	Priority	Normal

Actor:

- Super Admin
- Class admin
- Trainer

Summary:

- This use case allows user to create new syllabus.

Goal:

- Add new Syllabus

Triggers:

- User click on "Add Syllabus" button.

Preconditions:

- User must login with 3 roles "Super Admin, Class Admin, Trainer"

Post Conditions:

- Success: User can add new syllabus.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Syllabus" in the menu	Show a dropdown list
2	User click on "Create Syllabus"	Show a form for admin to input syllabus information
3	User input information of syllabus	System check validation [Exception 1]
4	Click "Save" button	System show popup

Alternative: N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1	"Topic Name" is required	Focus on "Topic Name" field. Show error message
2	"Version" is required	Focus on "Version" field. Show error message
3	"Attendee Number" is required	Focus on "Attendee Number" field. Show error message

4	"Technical Requirement" is required	Focus on "Technical Requirement" field. Show error message
5	"Course Objective" is required	Focus on "Course Objective" field. Show error message
6	"Delivery Principle" is required	Focus on "Delivery Principle" field. Show error message

Relationships: N/A

Business Rules: N/A

2.2.2.2. <Super admin, Class admin, Trainer> View Syllabus (UC_21)

Use Case Diagram

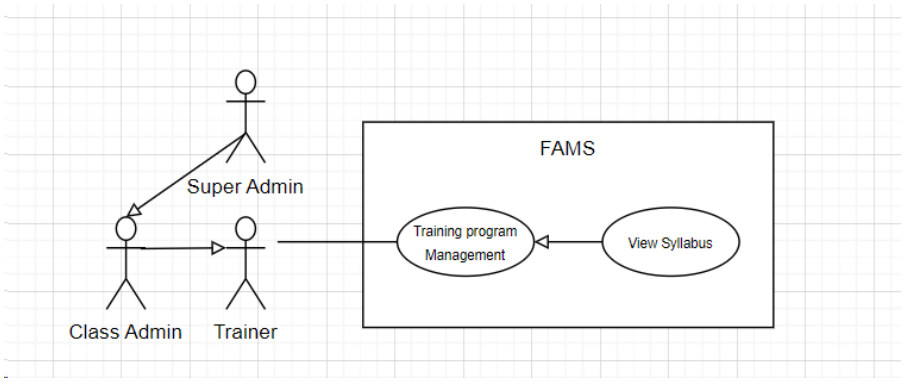


Figure 25 - <Super admin, Class admin, Trainer>View Syllabus

Use Case Specification

USE CASE – UC_21			
Use Case No.	**21**	**Use Case Version**	**1.0**
Use Case Name	**View Syllabus**		
Author	**Mã Ngọc Long Bảo**		
Date	**19/03/2024**	**Priority**	**Normal**
Actor: - Super Admin - Class admin - Trainer **Summary:**			

- This use case allows user to view information on syllabus.

Goal:

- User can view Syllabus's information

Triggers:

- User click on "View Syllabus" button.

Preconditions:

- User must login with 3 roles "Super Admin, Class Admin, Trainer"

Post Conditions:

- **Success:** User can view information on all syllabus.
- **Fail: N/A.**

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Syllabus at menu"	Show a dropdown list
2	User click on "View Syllabus" button.	Show information on all syllabus

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

User can see all syllabus's information

2.2.2.3. <Super admin, Class admin> Import Syllabus (UC_22)

Use Case Diagram

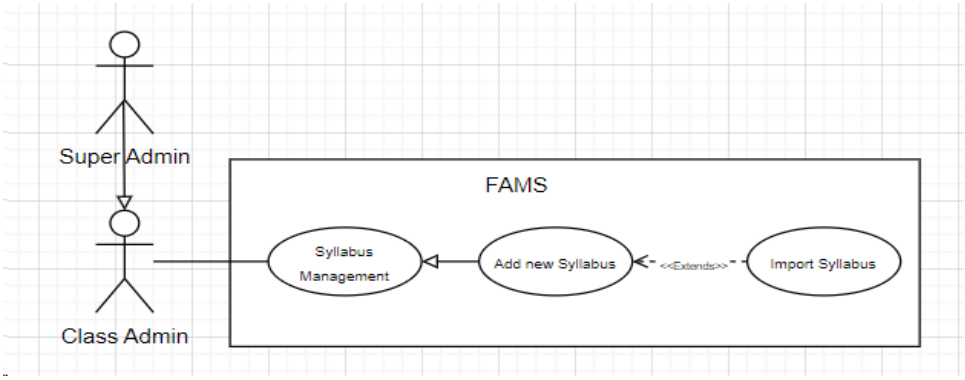


Figure 26 - <Super admin, Class admin> Import Syllabus

Use Case Specification

USE CASE – UC_22			
Use Case No.	22	**Use Case Version**	1.0
Use Case Name	Import Syllabus		
Author	Mã Ngọc Long Bảo		
Date	19/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class admin

Summary:

- This use case allows user to import new syllabus from excel or csv file.

Goal:

- Add new Syllabus

Triggers:

- User click on "Import Syllabus" button.

Preconditions:

- User must login with 2 roles "Super Admin, Class Admin"

Post Conditions:

- **Success:** User can add new syllabus.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Syllabus" in the menu	Show a dropdown list
2	User click on "Import"	Show a form for admin to import a file
4	User select choices of syllabus	
5	Click "Import" button	Show success message. [Exception 1]

Alternative: N/A

Exceptions: N/A

No	Cause	System Response

1	**Status is invalid**	**Focus on "Status" field. Show error message**
Relationships: N/A **Business Rules: N/A**		

2.2.2.4. <Super admin, Class admin, Trainer> Export syllabus list (UC_23)

Use Case Diagram

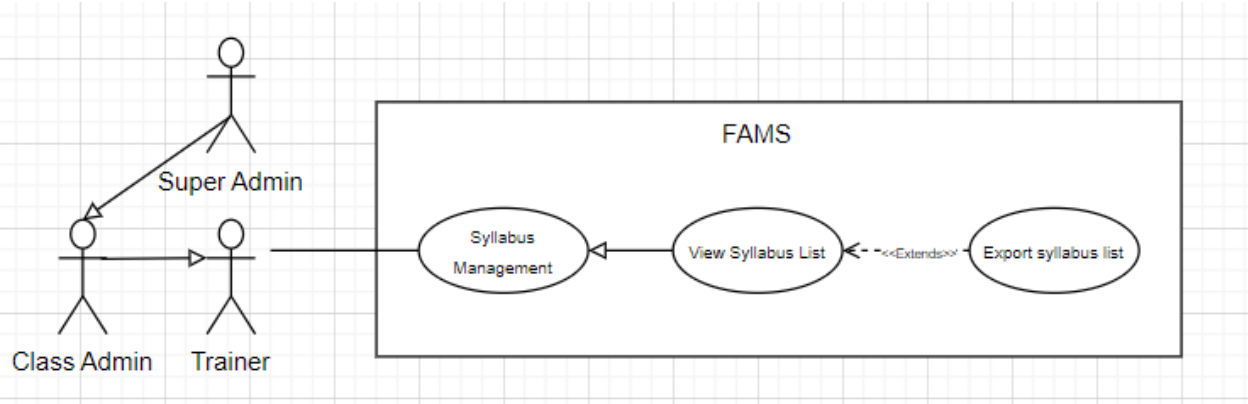


Figure 27 - <Super admin, Class admin, Trainer> Export syllabus list

Use Case Specification

USE CASE – UC_23			
Use Case No.	23	**Use Case Version**	1.0
Use Case Name	Export syllabus list		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal
Actor: - Super Admin - Class Admin - Trainer **Summary:** - This use case allows user to export all syllabus list **Goal:** - User can export all syllabus list as csv file. **Triggers:** - User clicks on "Export" label to download syllabus list **Preconditions:** - User must login with 3 role "Super Admin, Class Admin, Trainer" - System display success notification.			

Post Conditions:

- **Success:** Download successfully all syllabus list in csv file
- **Fail:** Can't export file

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Export" button	The system validates data and download to local storage

Alternative Scenario: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.2.5. <Super admin, Class admin, Trainer> Edit Syllabus (UC_24)

Use Case Diagram

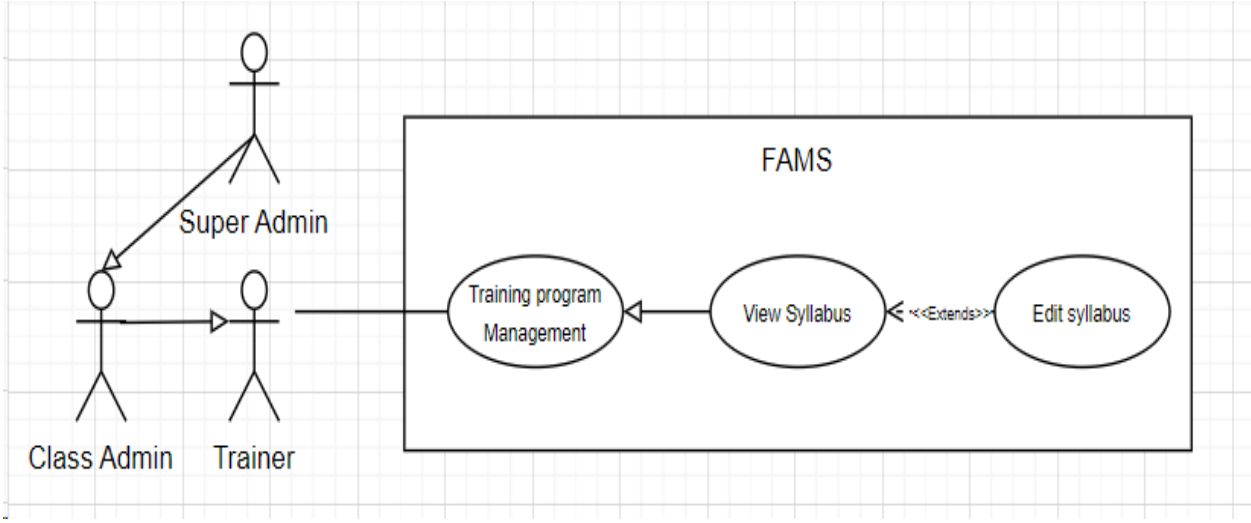


Figure 28 - <Super admin, Class admin, Trainer> Edit Syllabus

Use Case Specification

USE CASE – UC_24			
Use Case No.	**24**	**Use Case Version**	**1.0**
Use Case Name	**Edit syllabus**		
Author	**Mã Ngọc Long Bảo**		
Date	**19/03/2024**	**Priority**	**Normal**

Actor:

- Super Admin
- Class admin
- Trainer

Summary:

- This use case allows user to update syllabus.

Goal:

- Update Syllabus

Triggers:

- User click on "Edit Syllabus" button.

Preconditions:

- User must login with 3 roles "Super Admin, Class Admin, Trainer"

Post Conditions:

- Success: User can update syllabus.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Syllabus" in the menu	Show a dropdown list
2	User click on "View Syllabus" button.	Show information on all syllabus
3	User click dropdown button	Show a dropdown list
4	User click "Edit Syllabus"	Show a form for admin to input syllabus information
5	User input information of syllabus	System check validation [Exception 1]
6	Click "Save" button	System show popup

Alternative: N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1	"Topic Name" is required	Focus on "Topic Name" field. Show error message
2	"Version" is required	Focus on "Version" field. Show error message

3	"Attendee Number" is required	Focus on "Attendee Number" field. Show error message
4	"Technical Requirement" is required	Focus on "Technical Requirement" field. Show error message
5	"Course Objective" is required	Focus on "Course Objective" field. Show error message
6	"Delivery Principle" is required	Focus on "Delivery Principle" field. Show error message

Relationships: N/A

Business Rules: N/A

2.2.2.6. <Super admin, Class admin, Trainer> Search Syllabus (UC_25)

Use Case Diagram

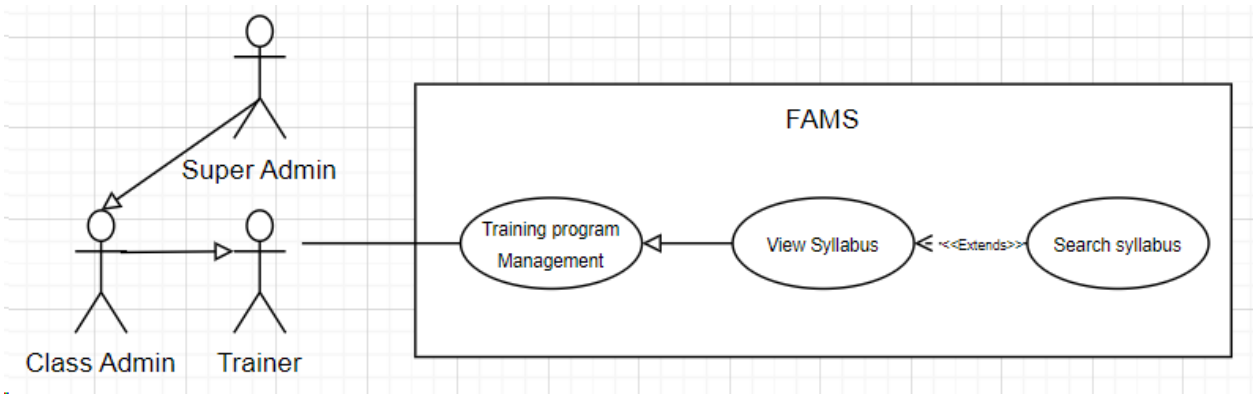


Figure 29 - <Super admin, Class admin, Trainer> Search Syllabus

Use Case Specification

USE CASE – UC_25			
Use Case No.	25	**Use Case Version**	1.0
Use Case Name	Search Syllabus		
Author	Mã Ngọc Long Bảo		
Date	19/03/2024	**Priority**	Normal
Actor: - Super Admin - Class admin - Trainer			

Summary:

- This use case allows user to search syllabus.

Goal:

- Update Syllabus successfully.

Triggers:

- User click on search bar.

Preconditions:

- User must login with 3 roles "Super Admin, Class Admin, Trainer"

Post Conditions:

- Success: User can search syllabus.
- Fail: N/A.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Syllabus" in the menu	Show a dropdown list
2	User click on "View Syllabus" button.	Show information on all syllabus
3	User click search bar	
4	User enter text to search	Return a search result

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.2.7. <Super admin, Class admin, Trainer> Delete Syllabus (UC_26)

Use Case Diagram

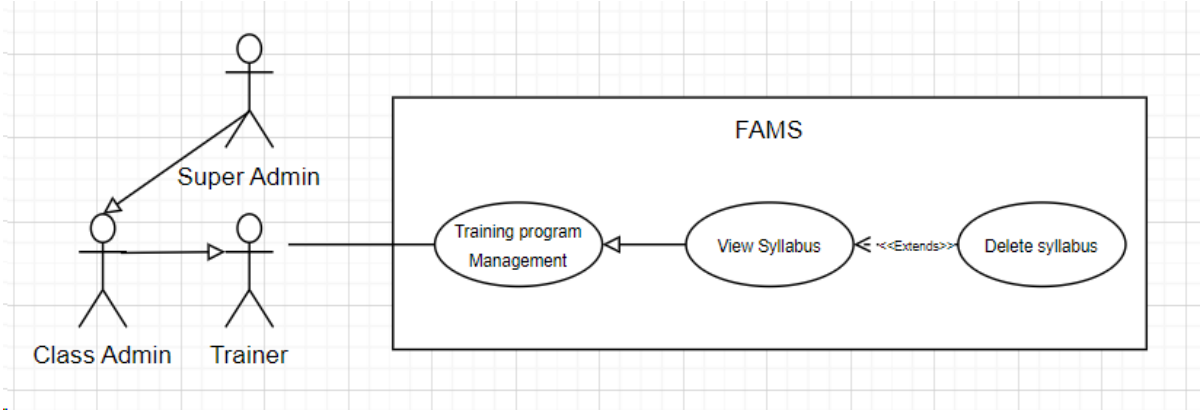


Figure 30 - <Super admin, Class admin, Trainer> Delete Syllabus

Use Case Specification

USE CASE – UC_26			
Use Case No.	26	**Use Case Version**	1.0
Use Case Name	Delete training program		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin

Summary:

- This use case allows user delete the training program

Goal:

- User can delete the training programs

Triggers:

- User clicks on "Delete program" button

Preconditions:

- User must login with 2 role "Super Admin, Class Admin"
- System display table list of training program.
- Program must be de-active.

Post Conditions:

- **Success:** Delete successfully and show
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response

1	User clicks on the three dots button at the last of training program in the list.	The system displays "Manage" modal: - Add training program - Edit syllabus - Duplicate syllabus - Delete syllabus
2	User presses "Delete syllabus" button.	The system deletes the syllabus.

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on "Cancel" button.	System goes back to "View syllabus" page.

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.2.8. <Super admin, Class admin, Trainer> De-activate Syllabus (UC_27)

Use Case Diagram

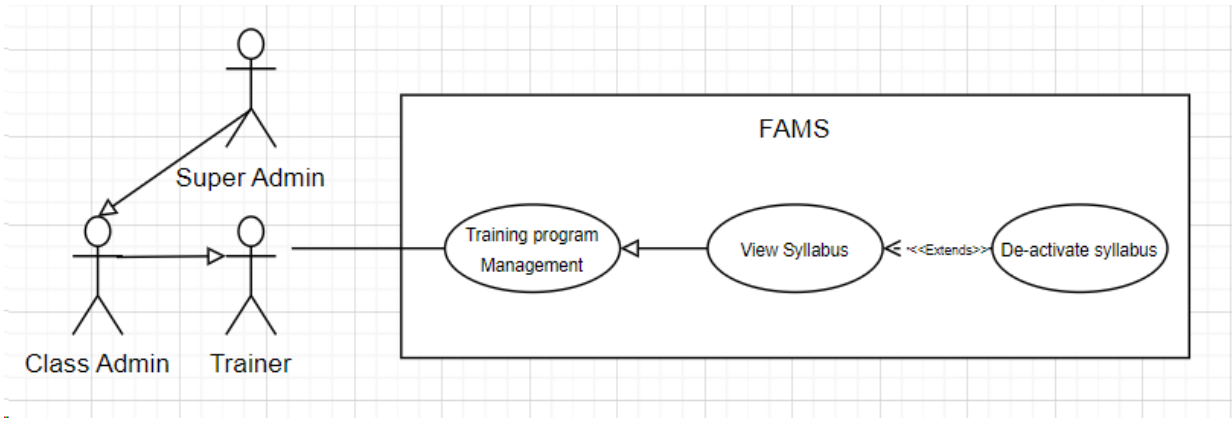


Figure 31 - <Super admin, Class admin, Trainer> De-activate Syllabus

Use Case Specification

USE CASE – UC_27			
Use Case No.	**27**	**Use Case Version**	**1.0**
Use Case Name	**De-activate Syllabus**		
Author	**Mã Ngọc Long Bảo**		

Date	19/03/2024	Priority	Normal

Actor:

- Super Admin
- Class admin
- Trainer

Summary:

- This use case allows user to de-activate syllabus.

Goal:

- De-activate Syllabus

Triggers:

- Admin click on "De-activate syllabus" button.

Preconditions:

- User must login with 3 roles "Super Admin, Class Admin, Trainer"

Post Conditions:

- Success: User can de-activate syllabus.
- Fail: N/A.

Main Success Scenario:

Step	Actor Action	System Response
1	User click "De-active syllabus" button.	The system saves new status of syllabus and display new list

Alternative: N/A

Exceptions: N/A

No	Cause	System Response
1	Status is invalid	Focus on "Status" field. Show error message

Relationships: N/A

Business Rules: N/A

2.2.2.9. <Super admin, Class admin, Trainer> Duplicate Syllabus (UC_28)

Use Case Diagram

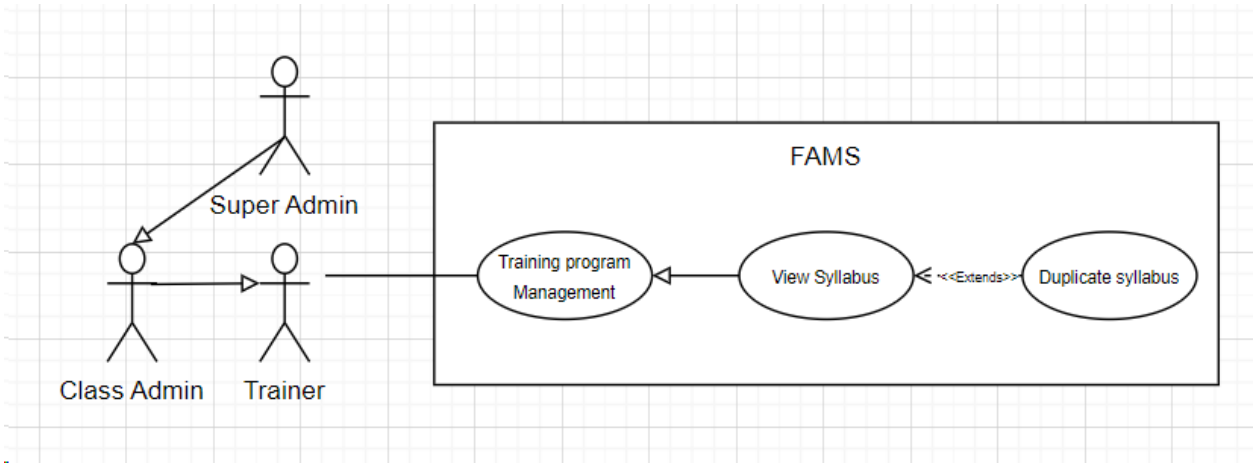


Figure 32 - <Super admin, Class admin, Trainer> Duplicate Syllabus

Use Case Specification

USE CASE – UC_28			
Use Case No.	28	**Use Case Version**	1.0
Use Case Name	Duplicate Syllabus		
Author	Mã Ngọc Long Bảo		
Date	19/03/2024	**Priority**	Normal

Actor:
- Super Admin
- Class admin
- Trainer

Summary:
- This use case allows user to duplicate syllabus.

Goal:
- Duplicate Syllabus

Triggers:
- User clicks on "Duplicate syllabus" button.

Preconditions:
- User must login with 3 roles "Super Admin, Class Admin, Trainer"

Post Conditions:
- Success: User can duplicate syllabus.
- Fail: N/A.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click "Duplicate syllabus" button	The system displays new duplicate syllabus.

Alternative: N/A

Exceptions:

No	Cause	System Response
1	Status is invalid	Focus on "Status" field. Show error message

Relationships: N/A

Business Rules: N/A

2.2.2.10. <Super admin, Class admin, Trainer> Save as draft (UC_29)

Use Case Diagram

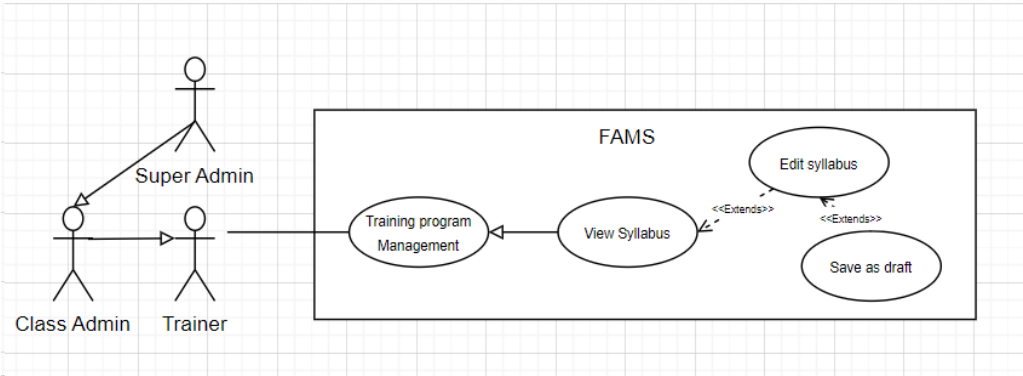


Figure 33 - <Super admin, Class admin, Trainer> Save as draft

Use Case Specification

USE CASE – UC_29			
Use Case No.	29	**Use Case Version**	1.0
Use Case Name	Save as draft		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin
- Trainer

Summary:

- This use case allows user to save as draft when edit or create syllabus

Goal:

- User can save as draft syllabus

Triggers:

- User clicks on "Save as draft" button.

Preconditions:

- User must login with 3 role "Super Admin, Class Admin, Trainer"

Post Conditions:

- **Success:** System display new syllabus in table list with status "draft"
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User edit or create new syllabus	The system display "Create syllabus" page
2	User click on "Save as draft" button	System save change and show new syllabus with status "draft"

Alternative Scenario:

No	Actor Action	System Response
1	User clicks on "Cancel" label	Go to "View syllabus" page
2	User clicks on "Back" button	Go back to previous step.

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.2.11. <Super admin, Class admin, Trainer> Search by Date picker (UC_30)

Use Case Diagram

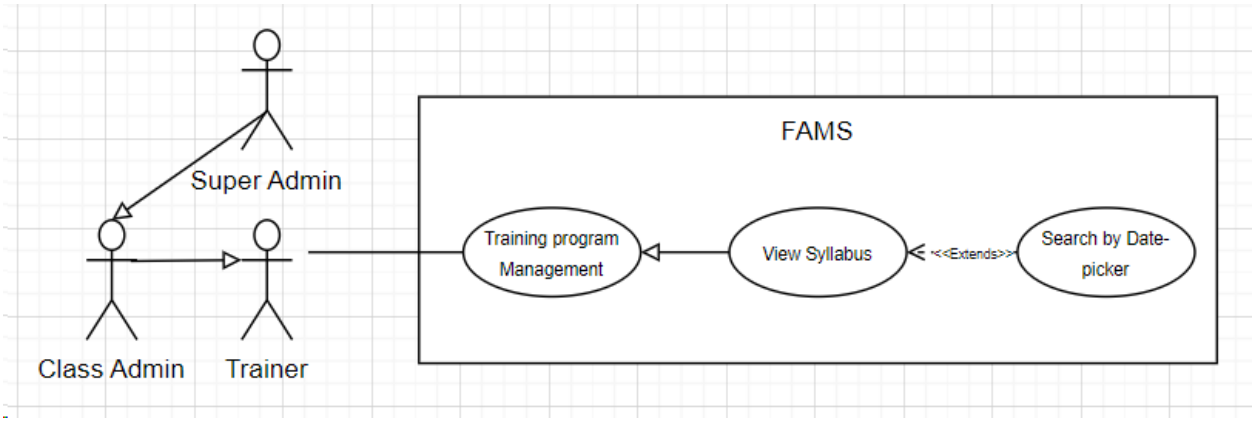


Figure 34 - <Super admin, Class admin, Trainer> Search by Date picker

Use Case Specification

USE CASE – UC_30			
Use Case No.	30	**Use Case Version**	1.0
Use Case Name	Search by Date picker		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin
- Trainer

Summary:

- This use case allows user to choose range of "date create by" and show list relate to what user choose

Goal:

- User can choose range of date by drag then drop date

Triggers:

- User clicks on "Created date" label then drag and drop from date to date to change date.

Preconditions:

- User must login with 3 role "Super Admin, Class Admin, Trainer"

Post Conditions:

- **Success:** Show new syllabus list which relate to what date user choose.
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Create date"	
2	User drag a day then drop click in another day to pick range of date	System shows new syllabus list

Alternative Scenario: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.2.12. <Super admin, Class admin, Trainer> Paginate Syllabus list (UC_31)

Use Case Diagram

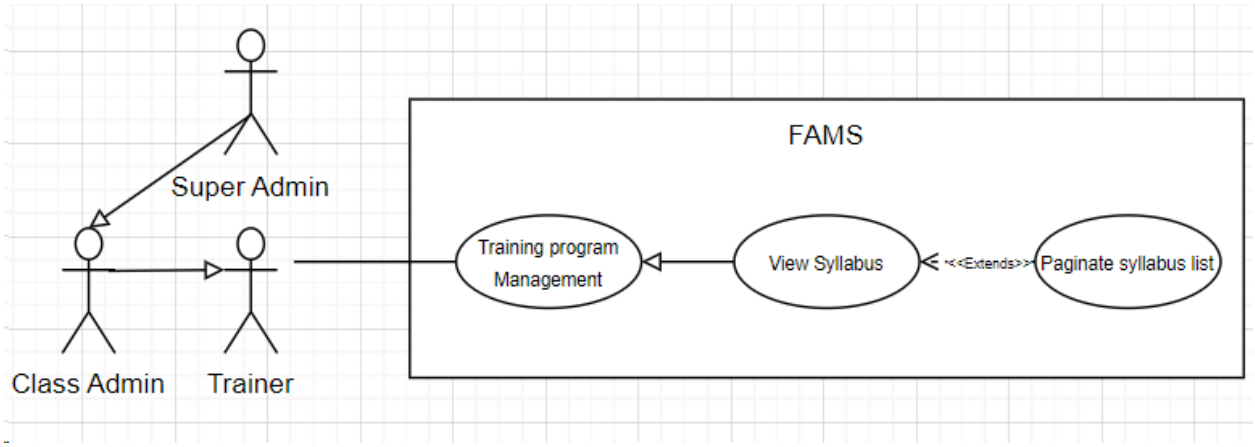


Figure 35 - <Super admin, Class admin, Trainer> Paginate Syllabus list

Use Case Specification

USE CASE – UC_31			
Use Case No.	31	**Use Case Version**	1.0
Use Case Name	Paginate syllabus list		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin
- Trainer

Summary:

- This use case allows user paginate the syllabus of table list in "View syllabus" page

Goal:

- User can paginate the syllabus list

Triggers:

- User clicks number or arrow on pagination bar.

Preconditions:

- User must login with 2 role "Super Admin, Class Admin"

Post Conditions:

- **Success:** Show results that correct what user choose on the pagination bar.

- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on number on pagination bar.	The list changes to show other syllabuses.

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on arrows on pagination bar.	The list changes to show other syllabuses.

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.2.13. <Super admin, Class admin, Trainer> Add unit training materials (UC_32)

Use Case Diagram

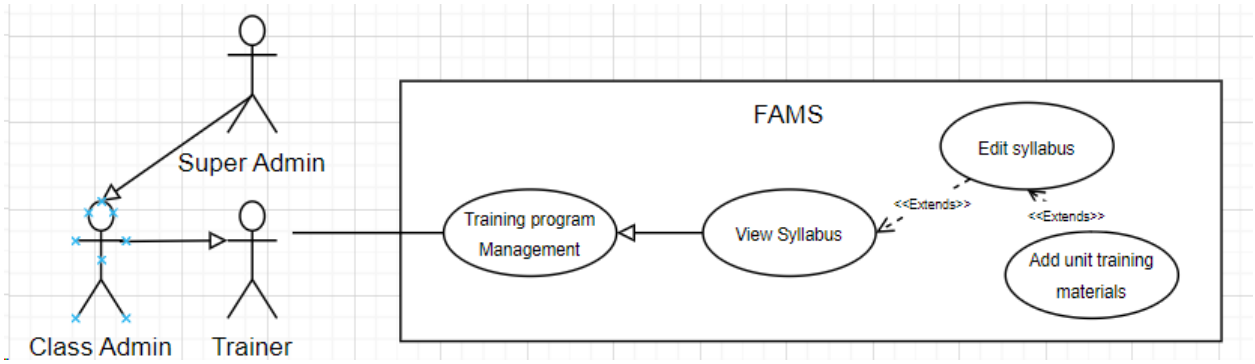


Figure 36 - <Super admin, Class admin, Trainer> Add unit training materials

Use Case Specification

USE CASE – UC_32			
Use Case No.	32	**Use Case Version**	1.0
Use Case Name	Add unit training materials		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin
- Trainer

Summary:

- This use case allows user to add new unit in a day of Outline field.

Goal:

- User can add new unit in a day.

Triggers:

- User clicks on "Add unit" in Outline field.

Preconditions:

- User must login with 3 role "Super Admin, Class Admin, Trainer"

Post Conditions:

- **Success:** System save and display successfully new unit chapter in Day field.
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Add unit"	The system request information in New Content form: - Name: Input required - Output Standard: Search required - Training time: Input required - Delivery type: Select one - Method: change status
2	User input data as the form	The system checks validate data
3	User click on "Save" button	The system displays new unit in Day field.

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on "Cancel" label.	Go back to Outline field.

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.2.14. <Super admin, Class admin, Trainer> Sort syllabus list (UC_33)

Use Case Diagram

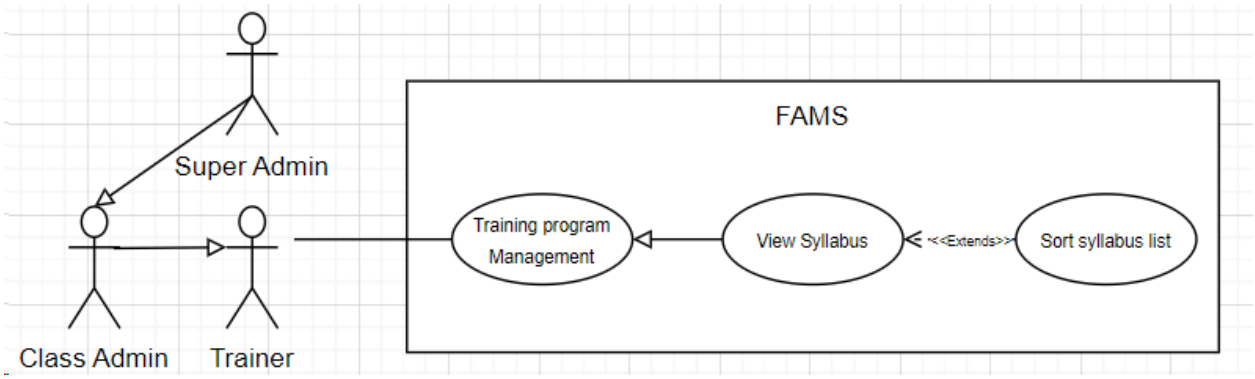


Figure 37 - <Super admin, Class admin, Trainer> Sort syllabus list

Use Case Specification

USE CASE – UC_33			
Use Case No.	33	**Use Case Version**	1.0
Use Case Name	Sort syllabus list		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin
- Trainer

Summary:

- This use case allows user sort the syllabus of table list in “View syllabus” page

Goal:

- User can sort the syllabus list

Triggers:

- User clicks on sort icon on the table list

Preconditions:

- User must login with 3 role “Super Admin, Class Admin, Trainer”

Post Conditions:

- **Success:** Show results that correct what user sort in the table list below.
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on sort icon.	The system displays field from A-Z

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.2.15. <Super admin, Class admin, Trainer> Show list per page (UC_34)

Use Case Diagram

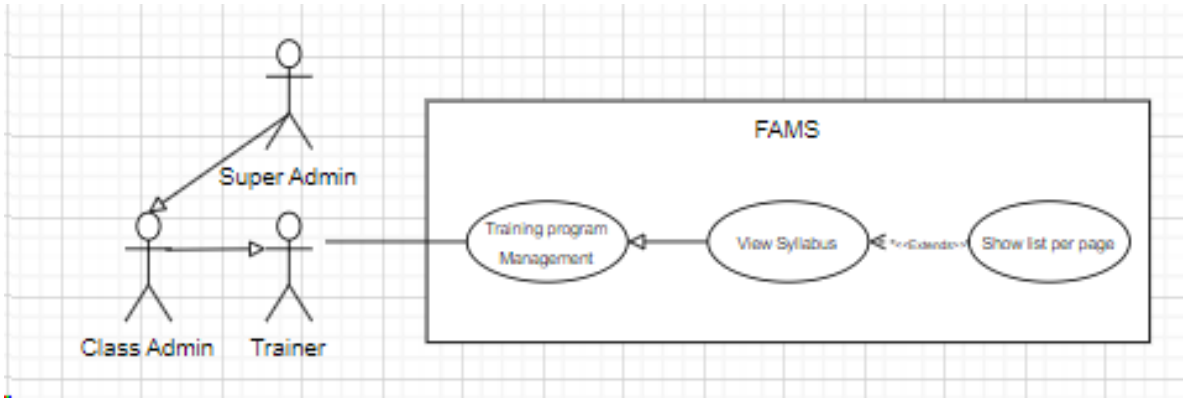


Figure 38 - <Super admin, Class admin, Trainer> Show list per page

Use Case Specification

USE CASE – UC_34			
Use Case No.	34	**Use Case Version**	1.0
Use Case Name	Show list per page		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin
- Trainer

Summary:

- This use case allows user to change number of programs show in one list page

Goal:

- User can change number of programs show in one list page.

Triggers:

- User clicks on "Show per page" label to choose number.

Preconditions:

- User must login with 3 role "Super Admin, Class Admin, Trainer"
- System display success table list of syllabuses

Post Conditions:

- **Success:** Show correct number program per 1 page in table list of syllabuses
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on "Show per page" label	The system show number to choose.
2	User choose number.	The system displays new syllabus table list

Alternative Scenario: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.2.16. <Super admin, Class admin> Download syllabus template (UC_35)

Use Case Diagram

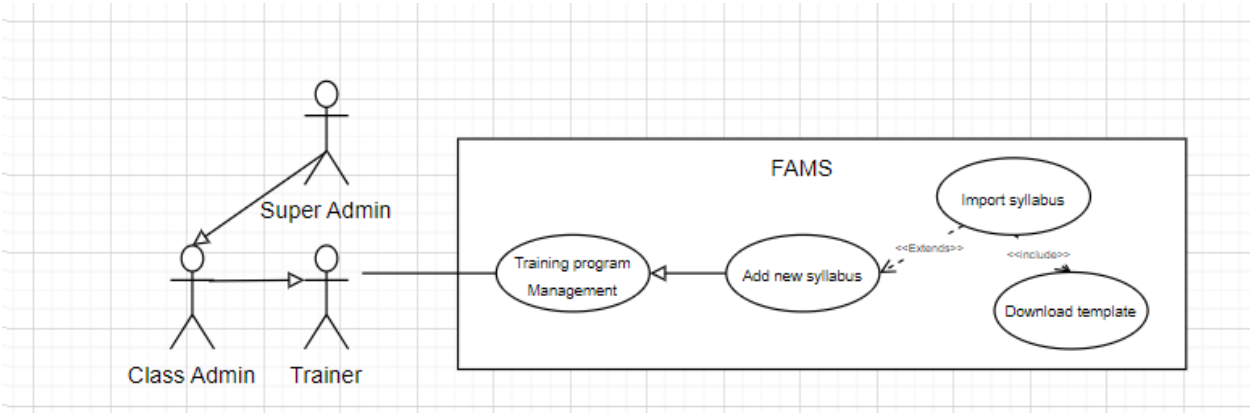


Figure 39 - < Super admin, Class admin> Download syllabus template

Use Case Specification

USE CASE – UC_35			
Use Case No.	35	**Use Case Version**	1.0
Use Case Name	Download syllabus template		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin

Summary:

- This use case allows user to download syllabus template

Goal:

- User can download syllabus template successfully.

Triggers:

- User clicks on "download" option in Import form.

Preconditions:

- User must login with 2 roles "Super Admin, Class Admin"

Post Conditions:

- **Success:** User can download the import template successfully.
- **Fail:** Show an error.

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on "download" option then save.	The system validates data and download to local storage

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on "Cancel" button.	System goes back to "Import" form.

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.3. Training Program Management Over View Use Case

Use Case Diagram

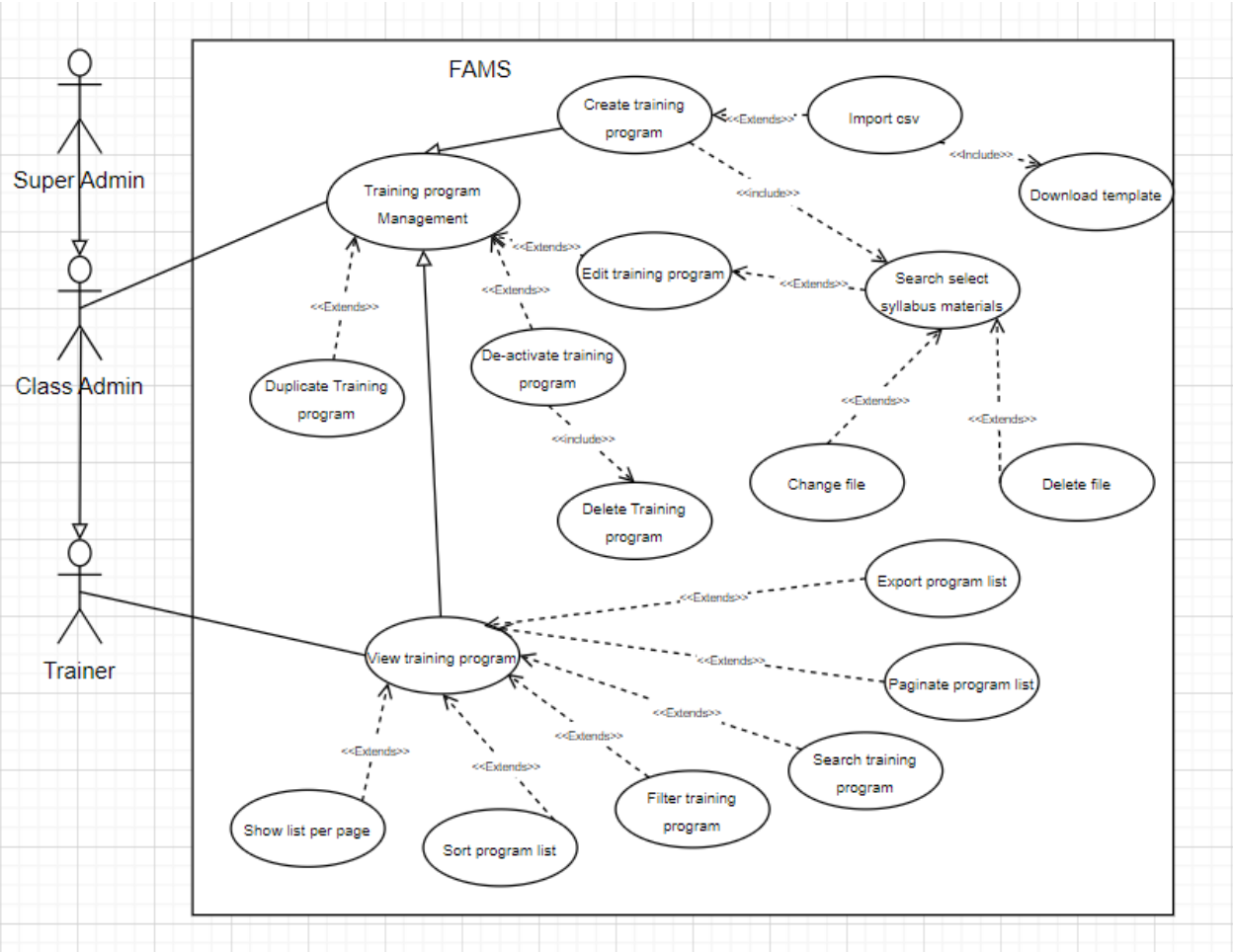


Figure 40 - Training Program Management Over View Use Case

2.2.3.1. <Super admin, Class admin, Trainer> Search training program (UC_36)

Use Case Diagram

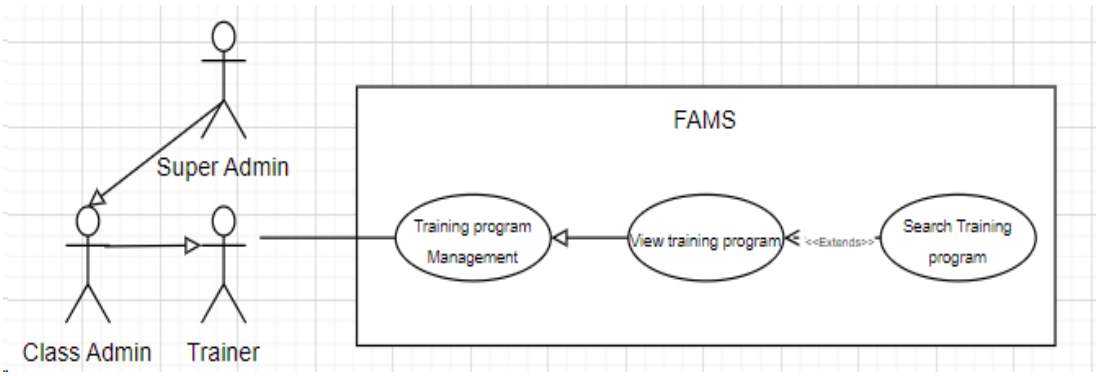


Figure 41 - <Super admin, Class admin, Trainer> Search training program

Use Case Specification

USE CASE – UC_36			
Use Case No.	36	**Use Case Version**	1.0
Use Case Name	Search training program		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin
- Trainer

Summary:

- This use case allows user search the training program in "View program"

Goal:

- User can search the training programs

Triggers:

- User clicks on "Search by…" label, type anything that relate about the program (name, id...) then press Enter.

Preconditions:

- User must login with 3 role "Super Admin, Class Admin, Trainer"

Post Conditions:

- **Success:** Add tags about what the user entered and show programs that correctly what user search in table list below.
- **Fail:** N/A.

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on "Search by…" label.	
2	User inputs information to search then press enter.	The system display tags every time the user presses enter and show the list program that relate to information in tags.

Alternative: N/A

Exceptions:

No	Cause	System Response
1	"Search by…" inputs must not be duplicated	Show error message "Please input other data".

Relationships: N/A

Business Rules:

- If user want to search by date must be in dd/mm/yyyy format.
- Data input must not same as others.

2.2.3.2. <Super admin, Class admin, Trainer> Filter training program (UC_36)

Use Case Diagram

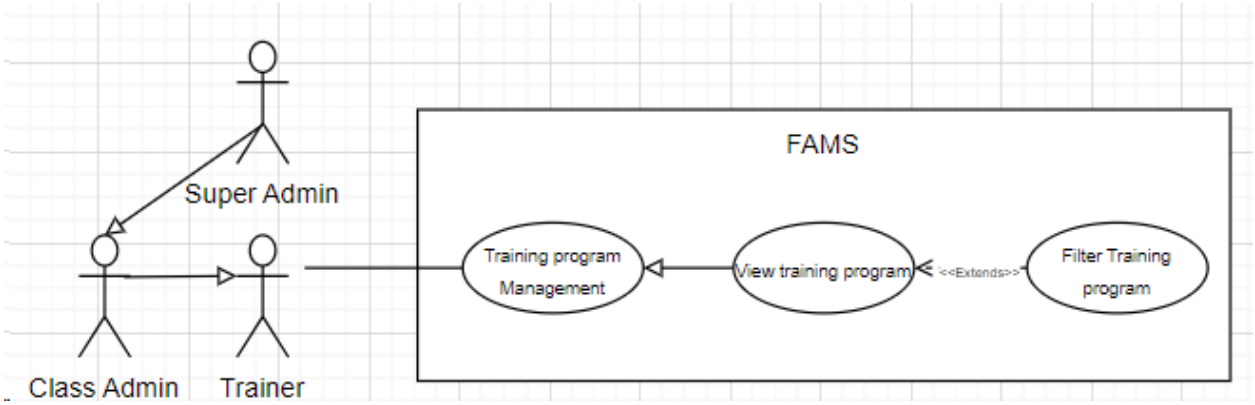


Figure 42 - <Super admin, Class admin, Trainer> Filter training program

Use Case Specification

USE CASE – UC_36			
Use Case No.	36	**Use Case Version**	1.0
Use Case Name	Filter training program		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin
- Trainer

Summary:

- This use case allows user filter the training program in "View program"

Goal:

- User can filter the training programs

Triggers:

- User clicks on "Filter" button

Preconditions:

- User must login with 3 role "Super Admin, Class Admin, Trainer"

Post Conditions:

- **Success:** Show results that correct what user inputs in the table list below.
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on "Filter" button.	The system request information in form: - : Free text input. [Exception 1]
2	User presses "Search" button.	The system displays all program relate to what user filter options.

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on "Cancel" button.	System goes back to "View program" page.

Exceptions:

No	Cause	System Response
1	No options to filter	Show error message "Please input at least 1 option".

Relationships: N/A

Business Rules:

- Fill at least 1 option in "Filter" form.

2.2.3.3. <Super admin, Class admin> Edit training program (UC_37)

Use Case Diagram

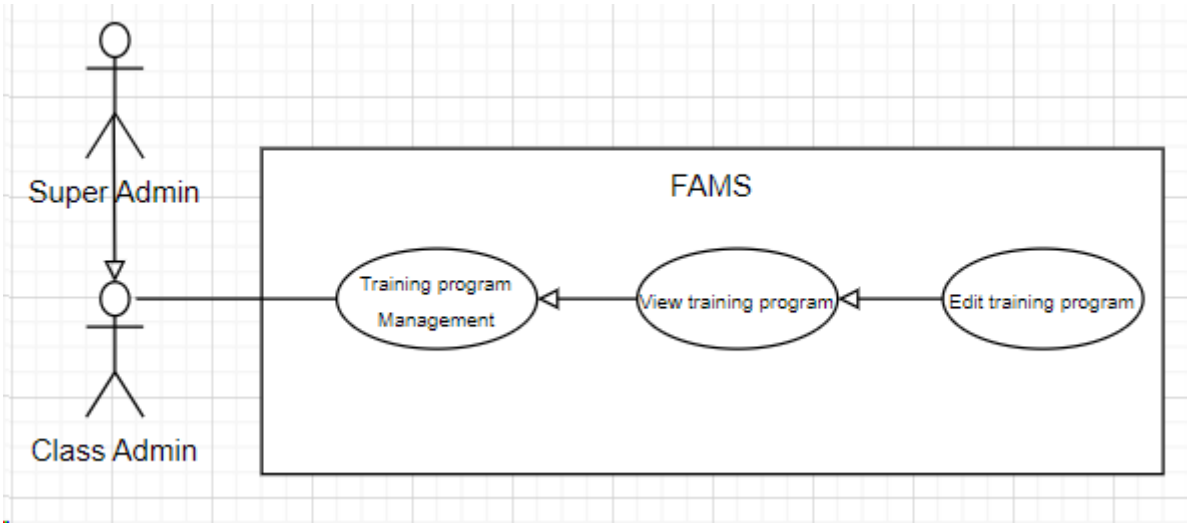


Figure 43 - <Super admin, Class admin> Edit training program

Use Case Specification

USE CASE – UC_37			
Use Case No.	37	**Use Case Version**	1.0
Use Case Name	Edit training program		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin

Summary:

- This use case allows user to edit the training program detail

Goal:

- User can edit the training programs detail.

Triggers:

- User clicks on "Edit program" option.

Preconditions:

- System display training program detail to edit.
- User must login with 2 role "Super Admin, Class Admin"

Post Conditions:

- **Success:** Edit training program successfully.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Edit program" option.	The system display training program details page to edit
2	User click on "Program name" label to edit the name then click "Save" to confirm.	The system changes the name of the program if not pass to next step.
3	User click on "Select Syllabus" label to add syllabus.	The system shows the dropdown list has result same as what user input.
4	User click on "Save" button	System saves new changes and shows success message.

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on "Back" button.	Go back to first step.

Exceptions:

No	Cause	System Response
1	New "Program name" must not duplicate	Show error message "Please input another name"

Relationships: N/A

Business Rules:

- "Program name" is not exceed 100 chars.

2.2.3.4. <Super admin, Class admin> Duplicate training program (UC_38)

Use Case Diagram

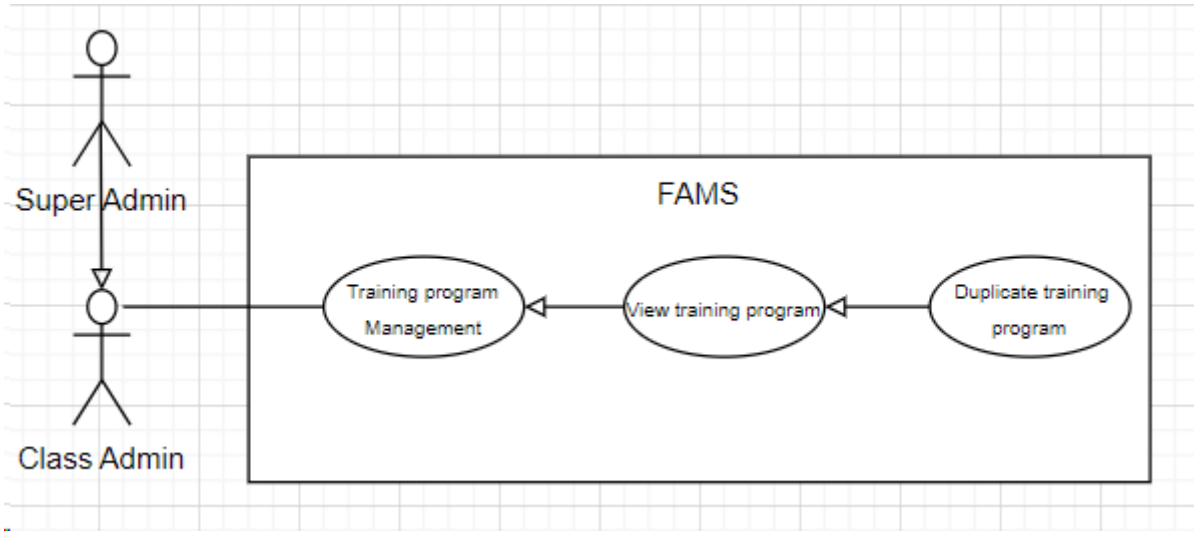


Figure 44 - <Super admin, Class admin> Duplicate training program

Use Case Specification

USE CASE – UC_38			
Use Case No.	38	**Use Case Version**	1.0
Use Case Name	Duplicate training program		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin

Summary:

- This use case allows user duplicate the training program.

Goal:

- User can duplicate the training programs

Triggers:

- User clicks on "Duplicate program" option.

Preconditions:

- User must login with 2 role "Super Admin, Class Admin"
- System display table list of training program.

Post Conditions:

- **Success:** Duplicate the training program successfully.
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response

1	User clicks on the three dots button at the last of training program in the list.	The system displays "Manage" form: - Training materials - Edit program - Duplicate program - De-activate program - Delete program
2	User presses "Duplicate program" option.	The system displays the program same as program that be duplicated

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on "Cancel" button.	System goes back to "View program" page.

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.3.5. <Super admin, Class admin> Delete training program (UC_39)

Use Case Diagram

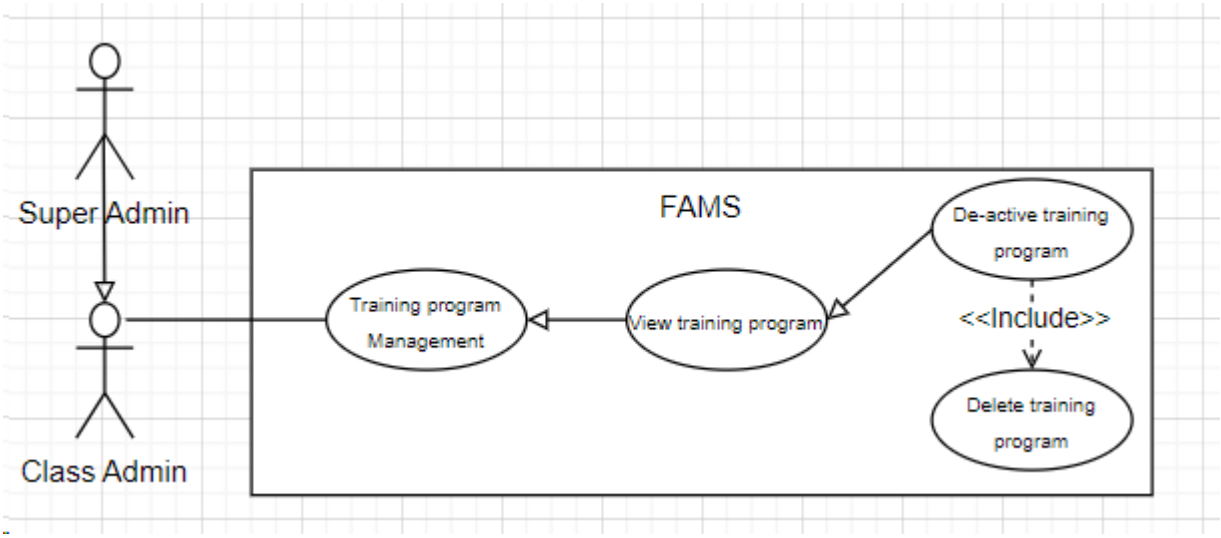


Figure 45 - <Super admin, Class admin> Duplicate training program

Use Case Specification

USE CASE – UC_39

Use Case No.	39	**Use Case Version**	1.0
Use Case Name	Delete training program		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin

Summary:

- This use case allows user delete the training program

Goal:

- User can delete the training programs

Triggers:

- User clicks on "Delete program" button

Preconditions:

- User must login with 2 role "Super Admin, Class Admin"
- System display table list of training program.
- Program must be de-active.

Post Conditions:

- **Success:** Delete successfully and show
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on the three dots button at the last of training program in the list.	The system displays "Manage" modal: - Training materials - Edit program - Duplicate program - De-activate program - Delete program
2	User presses "Delete program" button.	The system deletes the program.

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on "Cancel" button.	System goes back to "View program" page.

Exceptions: N/A **Relationships:** N/A **Business Rules:** N/A

2.2.3.6. <Super admin, Class admin> De-activate training program (UC_40)

Use Case Diagram

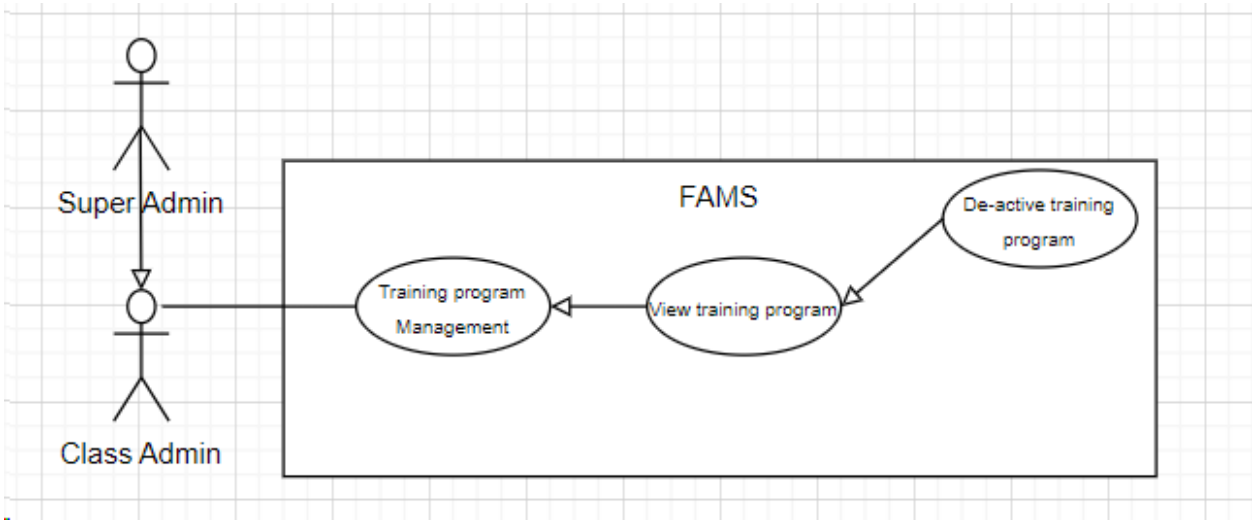


Figure 46 - <Super admin, Class admin> De-activate training program

Use Case Specification

USE CASE –UC_40			
Use Case No.	40	**Use Case Version**	1.0
Use Case Name	De-activate training program		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin

Summary:

- This use case allows user de-activate the training program.

Goal:

- User can de-activate the training programs.

Triggers:

- User clicks on "De-activate program" option.

Preconditions:

- User must login with 2 role "Super Admin, Class Admin"
- System display table list of training program.

Post Conditions:

- **Success:** Change status of the training program.
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on the three dots button at the last of training program in the list.	The system displays "Manage" modal: - Training materials - Edit program - Duplicate program - De-activate program - Delete program
2	User presses "De-activate program" button.	The system changes the status of the program

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on "Cancel" button.	System goes back to "View program" page.

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.3.7. <Super admin, Class admin>Import csv (UC_41)

Use Case Diagram

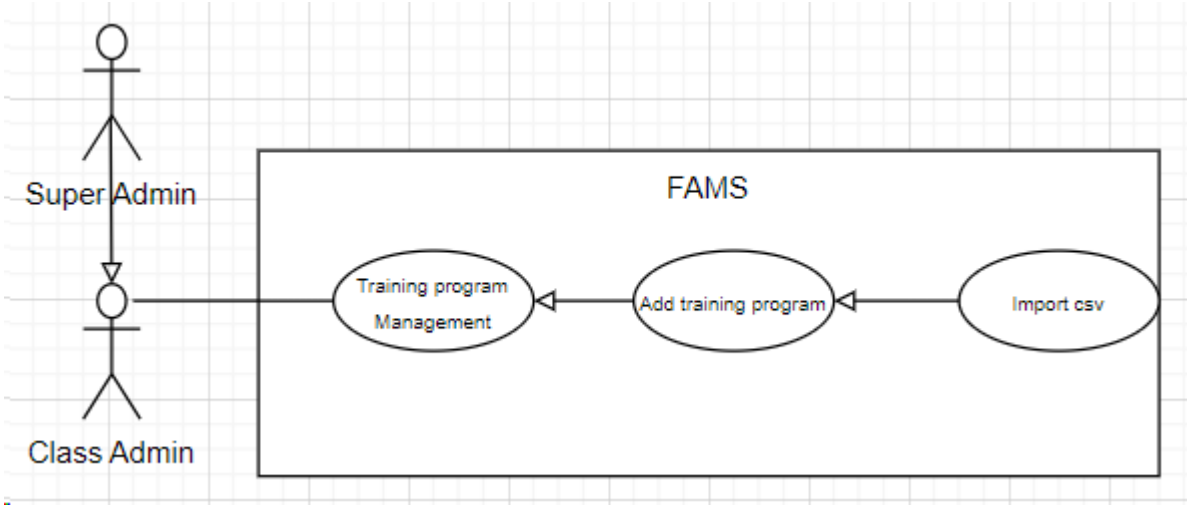


Figure 47 - <Super admin, Class admin>Import csv

Use Case Specification

USE CASE – UC_41			
Use Case No.	41	**Use Case Version**	1.0
Use Case Name	Import csv		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin

Summary:

- This use case allows user import new training program.

Goal:

- User can import the training programs

Triggers:

- User clicks on "Import" button

Preconditions:

- User must login with 2 role "Super Admin, Class Admin"

Post Conditions:

- **Success:** Add new training program success and it can be showed on training program list.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on “Import” button.	The system request information in Import form: Import Setting: - File(csv): excel or csv file required. - Encoding type: Select options - Column separator: Select options - Import template: Not required to click Duplicate Control: - Scanning: Select options - Duplicate handle: Select options
2	User input data then presses “Import” button in Import form	System validates the information and shows succesfull message.

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on “Cancel” button.	System goes back to “Import” form.

Exceptions 1:

No	Cause	System Response
1	“File(csv)” ís empty	Focus in “File(csv)” field. Show error message
2	“Encoding type” ís not selected	Focus in “Encoding type“ field. Show error message
3	“Column separator” ís not selected	Focus in “Column separator” field. Show error message
4	“Scanning” ís not selected	Focus in “Scanning” field. Show error message
5	“Duplicate handle” ís not selected	Focus in “Duplicate handle” field. Show error message

Exceptions 2:

No	Cause	System Response
1	“File(csv)” ís wrong type	Show error message

Relationships: N/A

Business Rules:

- File must be excel or csv type.

2.2.3.8. <Super admin, Class admin, Trainer> Export training program (UC_42)

Use Case Diagram

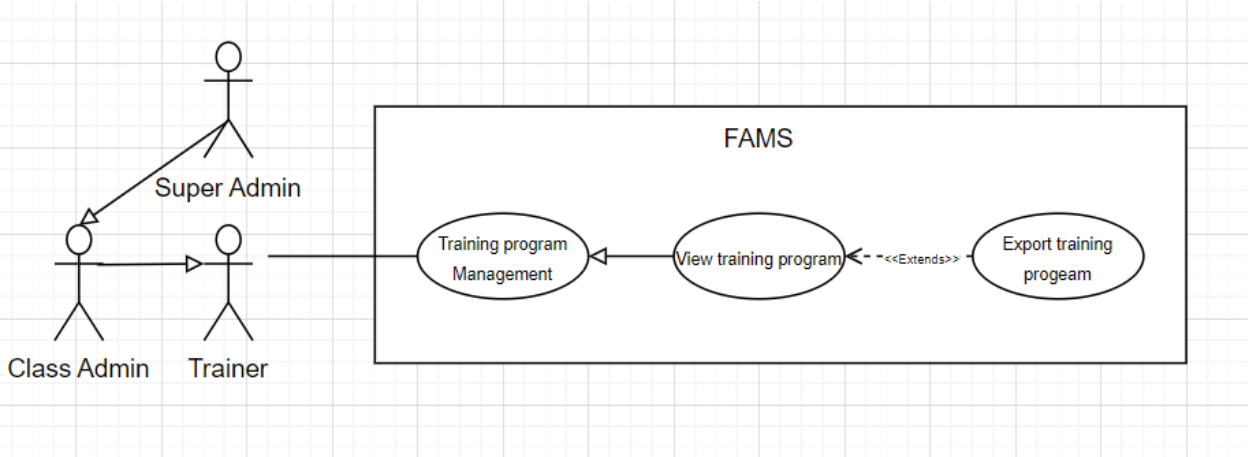


Figure 48 - <Super admin, Class admin, Trainer> Export training program

Use Case Specification

USE CASE – UC_42			
Use Case No.	42	**Use Case Version**	1.0
Use Case Name	Export training program		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin
- Trainer

Summary:

- This use case allows user to export all training programs

Goal:

- User can export all training program as csv file.

Triggers:

- User clicks on "Export" label to download training program list

Preconditions:

- User must login with 3 role "Super Admin, Class Admin, Trainer"
- System display success notification.

Post Conditions:

- **Success:** Download successfully all training program list in csv file
- **Fail:** Can't export file

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Export" button	The system validates data and download to local storage

Alternative Scenario: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.3.9. <Super admin, Class admin, Trainer>Sort training program list (UC_43)

Use Case Diagram

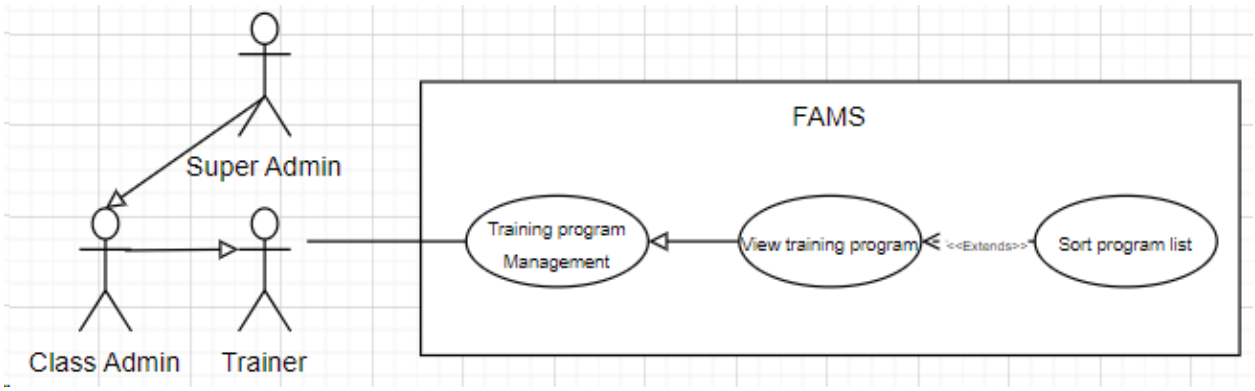


Figure 49 - <Super admin, Class admin, Trainer>Sort training program list

Use Case Specification

USE CASE – UC_43			
Use Case No.	43	**Use Case Version**	1.0
Use Case Name	Sort training program list		
Author	Mai Tiến Hoàng		

Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin
- Trainer

Summary:

- This use case allows user sort the training program of table list in "View program" page

Goal:

- User can sort the training program list

Triggers:

- User clicks on sort icon on the table list

Preconditions:

- User must login with 3 role "Super Admin, Class Admin, Trainer"

Post Conditions:

- **Success:** Show results that correct what user sort in the table list below.
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on sort icon.	The system displays field from A-Z

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.3.10. <Super admin, Class admin, Trainer>Paginate training program list (UC_44)

Use Case Diagram

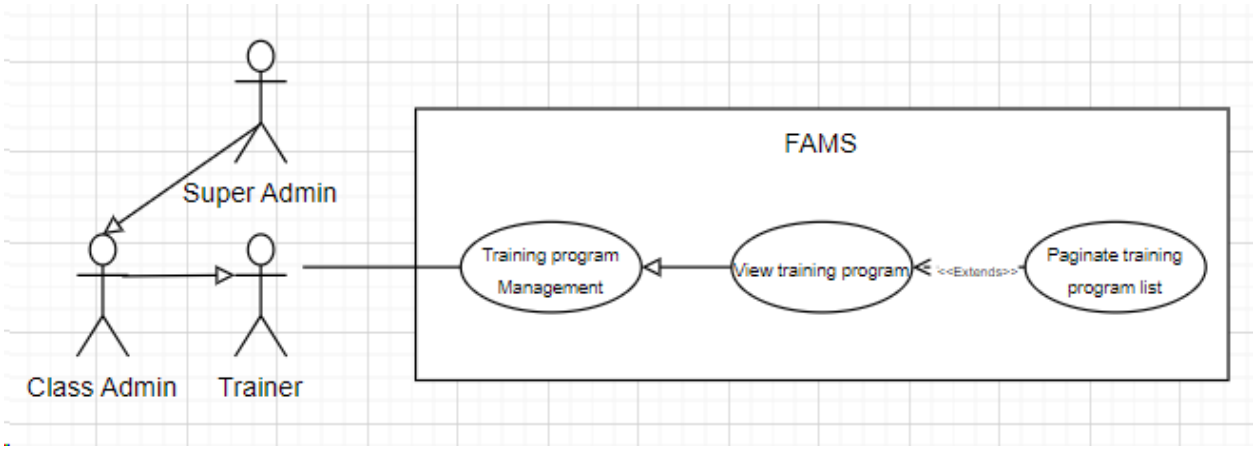


Figure 50 - <Super admin, Class admin, Trainer>Paginate training program list

Use Case Specification

USE CASE – UC_44			
Use Case No.	44	**Use Case Version**	1.0
Use Case Name	Paginate training program list		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin
- Trainer

Summary:

- This use case allows user paginate the training program of table list in "View program" page

Goal:

- User can paginate the training program list

Triggers:

- User clicks number or arrow on pagination bar.

Preconditions:

- User must login with 2 role "Super Admin, Class Admin"

Post Conditions:

- **Success:** Show results that correct what user choose on the pagination bar.
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on number on pagination bar.	The list changes to show other programs.

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on arrows on pagination bar.	The list changes to show other programs.

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.3.11. <Super admin, Class admin>Download training program template (UC_45)

Use Case Diagram

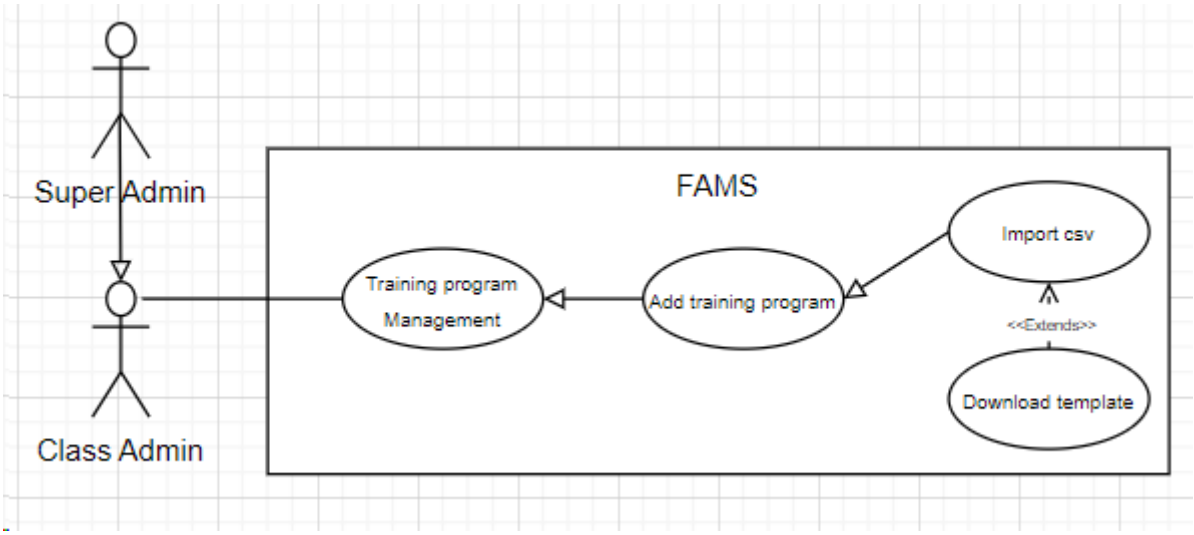


Figure 51 - <Super admin, Class admin>Download training program template

Use Case Specification

USE CASE – UC_45			
Use Case No.	45	**Use Case Version**	1.0
Use Case Name	Download training program template		

Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin

Summary:

- This use case allows user download the import template

Goal:

- User can download the import template successfully.

Triggers:

- User clicks on "download" option in Import form.

Preconditions:

- User must login with 2 role "Super Admin, Class Admin"

Post Conditions:

- **Success:** User can download the import template successfully.
- **Fail: Show an error.**

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on "download" option.	The system validates data and download to local storage

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on "Cancel" button.	System goes back to "Import" form.

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.3.12. <Super admin, Class admin>Search syllabus training materials (UC_46)

Use Case Diagram

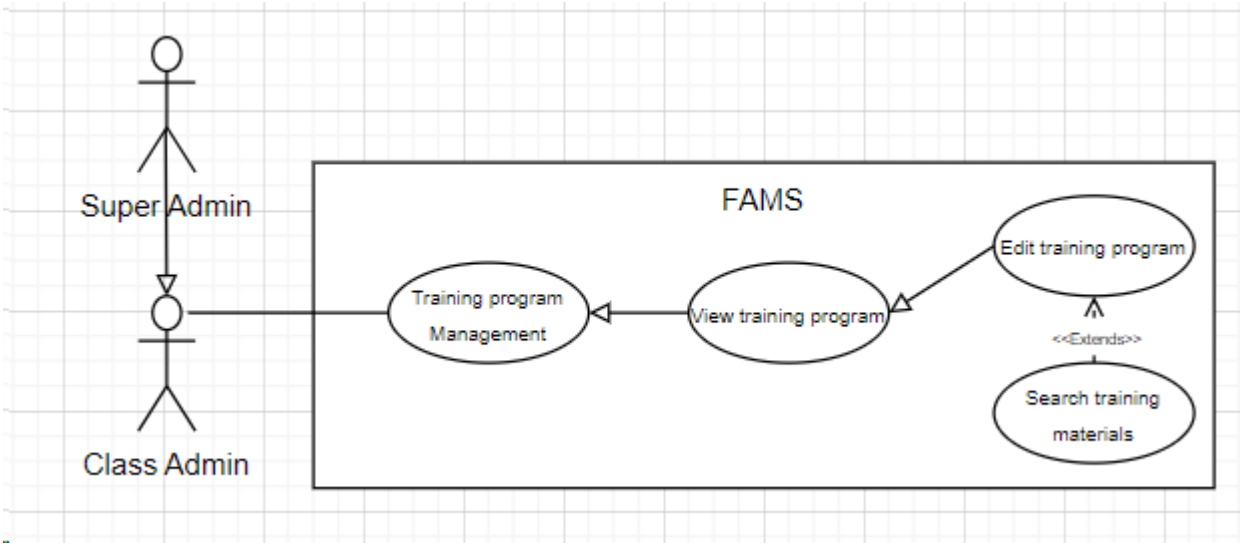


Figure 52 - <Super admin, Class admin>Search syllabus training materials

Use Case Specification

USE CASE – UC_46			
Use Case No.	46	**Use Case Version**	1.0
Use Case Name	Search syllabus training materials		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal
Actor: - Super Admin - Class Admin **Summary:** - This use case allows user to search syllabus training materials **Goal:** - User can search syllabus training materials. **Triggers:** - User clicks on “Select Syllabus” label, type data then choose the relate. **Preconditions:** - User must login with 2 role “Super Admin, Class Admin” **Post Conditions:** - **Success:** System show successfully result.			

- **Fail: Show an error.**

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on "Select Syllabus" label.	
2	User input data then choose the relate.	The system display content relate to what user search.

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on "X" icon.	The system deletes the content has just been searched.

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.1.12. <Super admin, Class admin>Change training materials file (UC_47)

Use Case Diagram

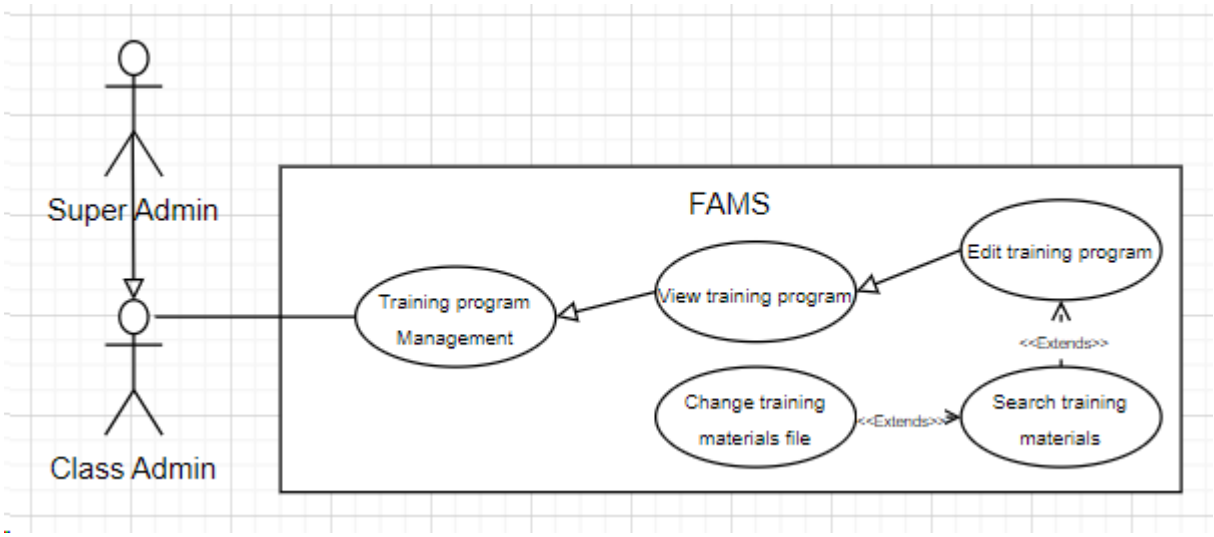


Figure 53 - <Super admin, Class admin>Change training materials file

Use Case Specification

USE CASE – UC_47			
Use Case No.	47	**Use Case Version**	1.0
Use Case Name	Change training materials file		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin

Summary:

- This use case allows user change the training materials file

Goal:

- User can change the training materials file

Triggers:

- User clicks on pen icon in training materials form

Preconditions:

- User must login with 2 role "Super Admin, Class Admin"

Post Conditions:

- Success: Change training materials file successfully
- Fail: N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on the three dots button at the last of training program in the list.	The system displays "Manage" modal: - Training materials - Edit program - Duplicate program - De-activate program - Delete program
2	User clicks on ""	
3	User clicks on "Training program" button.	The system go to the "Training program" form

4	User presses pen icon	The system deletes the materials

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on "Cancel" button.	System goes back to "View program" page.

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.3.13. <Super admin, Class admin>Delete training program materials (UC_48)

Use Case Diagram

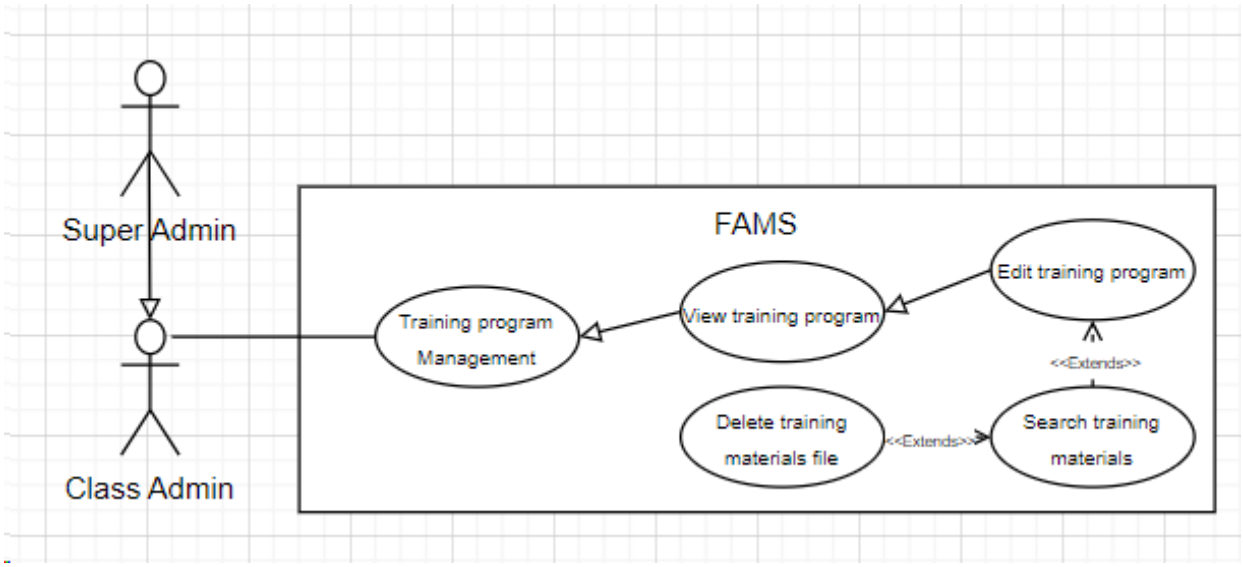


Figure 54 - <Super admin, Class admin>Delete training program materials

Use Case Specification

USE CASE – UC_48			
Use Case No.	48	**Use Case Version**	1.0
Use Case Name	Delete training materials		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin

Summary:

- This use case allows user delete the training materials

Goal:

- User can delete the training materials

Triggers:

- User clicks on delete icon in training materials form

Preconditions:

- User must login with 2 role "Super Admin, Class Admin"

Post Conditions:

- Success: Delete training materials successfully
- Fail: N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on the three dots button at the last of training program in the list.	The system displays "Manage" modal: - Training materials - Edit program - Duplicate program - De-activate program - Delete program
2	User clicks on "Training program" button.	The system goes to the "Training program" form
3	User presses delete icon	The system deletes the materials

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on "Cancel" button.	System goes back to "Training materials" form

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.3.14. <Super admin, Class admin>Create training program (UC_49)

Use Case Diagram

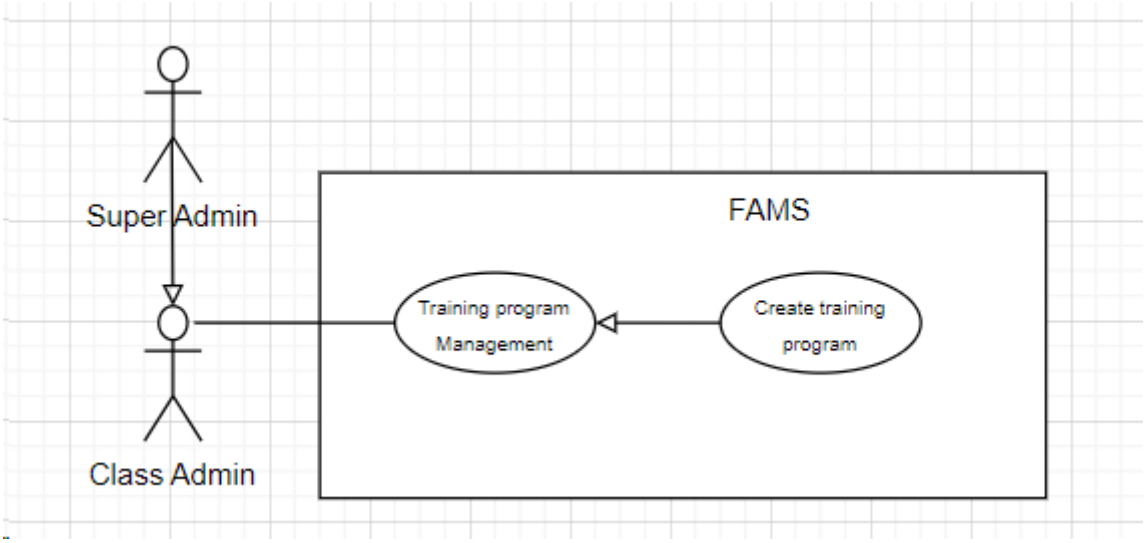


Figure 55 - <Super admin, Class admin>Create training program

Use Case Specification

USE CASE – UC_49			
Use Case No.	49	**Use Case Version**	1.0
Use Case Name	Create training program		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin

Summary:

- This use case allows user to create new training program**.**

Goal:

- User can create new training program.

Triggers:

- User clicks on "Add new" button or clicks on "Create program" button on side bar

Preconditions:

- User must login with 2 role "Super Admin, Class Admin"

Post Conditions:

- Success: Save new training program successfully
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on "Add new"	The system displays "Create training program" page
2	User click on "Program name" label to edit the name then click "Save" to confirm.	The system changes the name of the program if not pass to next step.
3	User click on "Select Syllabus" label to add syllabus.	The system shows the dropdown list has result same as what user input.
4	User click on "Save" button	System saves new changes and shows success message.

Alternative Scenario:

Step	Actor Action	System Response

1	User clicks on “Create program” button.	System goes back to “Training materials” form

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.3.15. <Super admin, Class admin, Trainer>Show list per page (UC_50)

Use Case Diagram

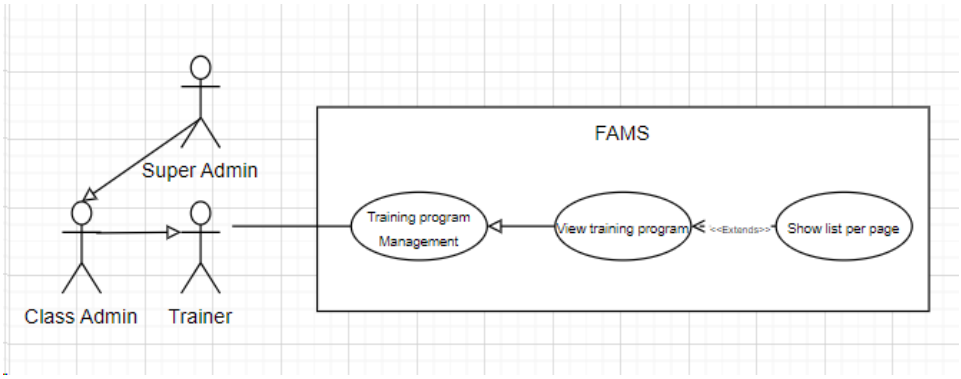


Figure 56 - <Super admin, Class admin, Trainer>Show list per page

Use Case Specification

USE CASE – UC_50			
Use Case No.	50	**Use Case Version**	1.0
Use Case Name	Show list per page		
Author	Mai Tiến Hoàng		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin
- Class Admin
- Trainer

Summary:

- This use case allows user to change number of programs show in one list page

Goal:

- User can change number of programs show in one list page.

Triggers:

- User clicks on “Show per page” label to choose number.

Preconditions:

- User must login with 3 role "Super Admin, Class Admin, Trainer"
- System display success table list of training program

Post Conditions:

- **Success:** Show correct number program per 1 page in table list of training program
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on "Show per page" label	The system show number to choose.
2	User choose number.	The system displays new training programs table list

Alternative Scenario: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.4. Class Management Over View Use Case

Use Case Diagram

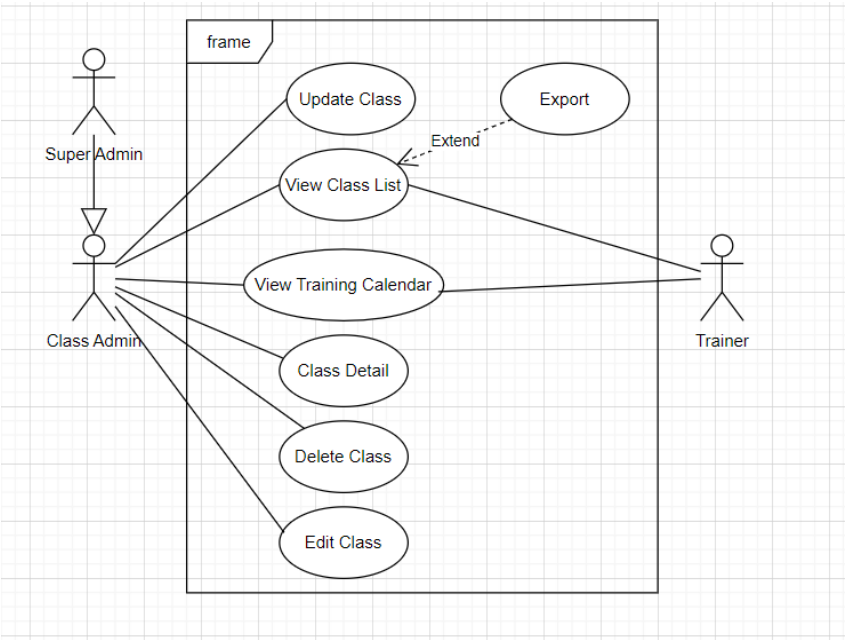


Figure 57 - Class Management Over View Use Case

2.2.3.1. <Super Admin, Class Admin, Trainer> View Class Detail (UC_51)

Use Case Diagram

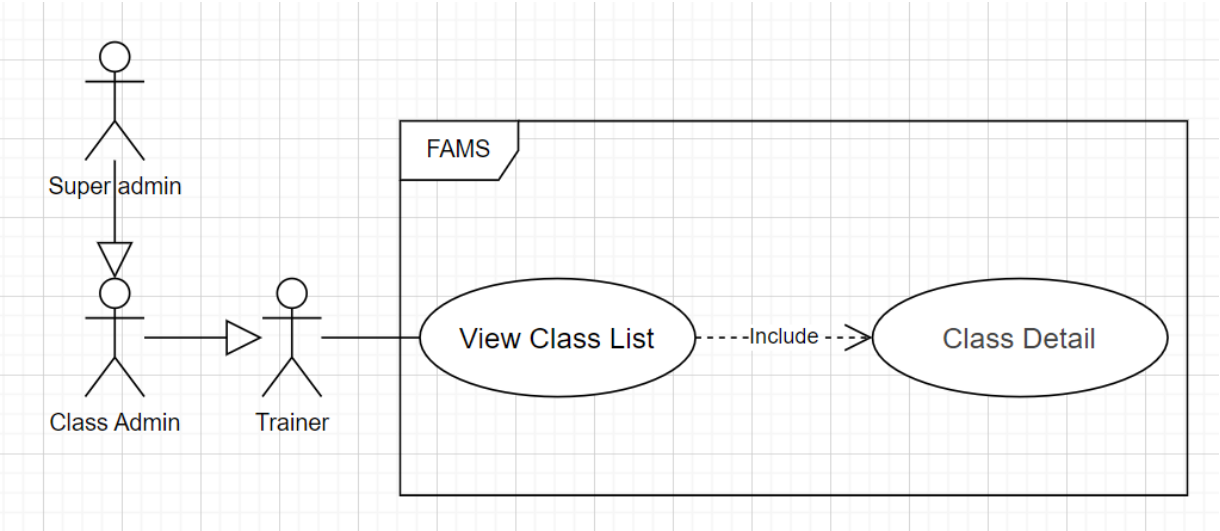


Figure 58 - <Super Admin, Class Admin, Trainer> View Class Detail Use Case

Use Case Specification

USE CASE – UC_51			
Use Case No.	51	**Use Case Version**	2.0
Use Case Name	View Class Detail		
Author	Mai Văn Quốc Tịnh		
Date	18/03/2024	**Priority**	Normal

Actor:

- Class Admin, Super Admin, Trainer.

Summary:

- This use case allows the admin, super admin, trainer to view details of class such as training program, list of candidates, trainer, list of syllabuses.

Goal:

- To allow the admin, super admin, trainer to view detailed information about a specific class within the system.

Triggers:

- Class admin, super admin, trainer click on the “name of class” in the view list class.

Preconditions:

- class admin, super admin, trainer at “class detail” page.

Post Conditions:

- Success: Users go to the Login page with their account and password have been registered.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Class admin, super admin, trainer view list of class selects a class from the list to view its details.	The system retrieves and displays information about the class list, including: +Class Name +Class Time +Fsu +Location +Training program +Syllabus +Calendar
2	Class admin, super admin, and trainers rate the displayed class list.	The system presents the class list clearly.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Class Descriptions: Detailed descriptions of the class content.

2.2.3.2. <Super Admin, Class Admin, Trainer> Search Class on List (UC_52)

Use Case Diagram

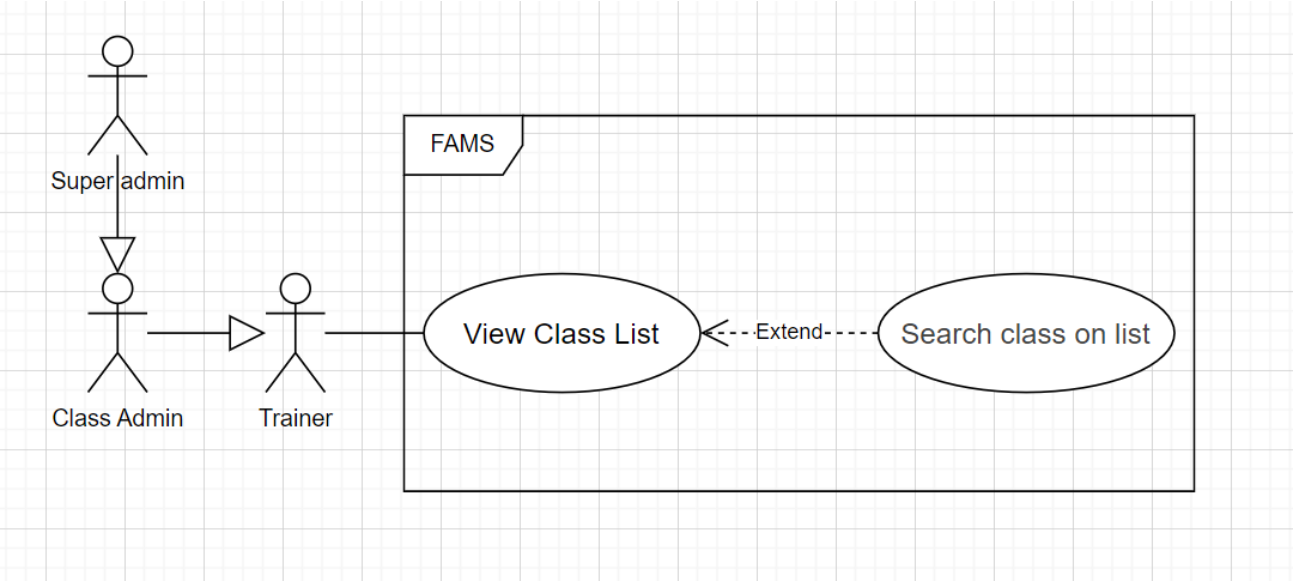


Figure 59 - <Super Admin, Class Admin, Trainer> Search Class on List Use Case

Use Case Specification

USE CASE – UC_52			
Use Case No.	52	**Use Case Version**	2.0
Use Case Name	Search Class on List		
Author	Mai Văn Quốc Tịnh		
Date	18/03/2024	**Priority**	Normal

Actor:

- Class Admin, Super Admin, Trainer.

Summary:

- This use case allows the admin, super admin, trainer to search .

Goal:

- To allows the admin, super admin, trainer to search for classes more easily and quickly

Triggers:

- Class admin, super admin, trainer click on the “Class” inside the bar and select “View Class”.

Preconditions:

- Class admin, super admin, trainer at “list of class” page.

Post Conditions:

- Success: Users go to the Login page with their account and password have been registered.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Class admin, super admin, trainer click on the search input you want to find	The system will return relevant classes based on the entered keywords

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Search Results: The system should display search results in a clear and organized manner, presenting relevant class details to users.
- Keyword Matching: The search functionality should match the entered keywords with the corresponding attributes of the classes accurately to ensure relevant results.

2.2.3.3. <Super Admin, Class Admin> Create Class (UC_53)

Use Case Diagram

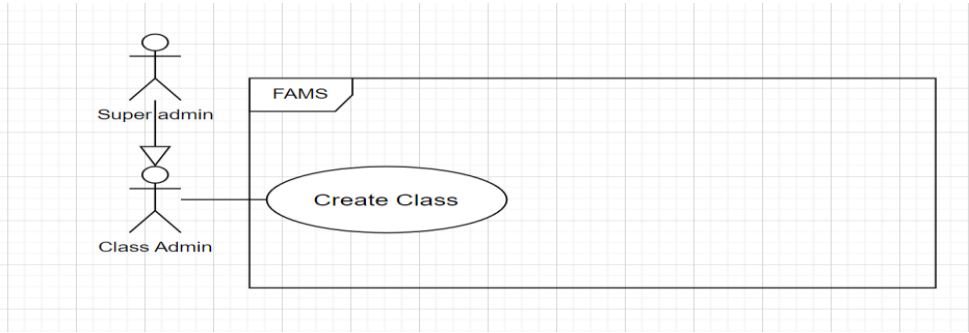


Figure 60 - <Super Admin, Class Admin> Create Class Use Case

Use Case Specification

USE CASE – UC_53			
Use Case No.	53	**Use Case Version**	2.0
Use Case Name	Create Class		
Author	Mai Văn Quốc Tịnh		
Date	18/03/2024	**Priority**	Normal
Actor: - Class Admin, Super Admin.			

Summary:

- Class admins and super admins can create classes by filling out a form with class details. The system validates inputs and provides error messages if necessary. Upon successful submission, the class is created, and users receive confirmation.

Goal:

- To enable admins and super admins to efficiently create new classes within the system by providing necessary class details, validating inputs, and confirming successful creation, facilitating effective management of class offerings.

Triggers:

- Class admin, super admin click on the "Class" inside the bar and select "Create Class".

Preconditions:

- Class admin, super admin at the "Create Class" page.

Post Conditions:

- Success: Users go to the Login page with their account and password have been registered.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Class admins, super admins click on the bar to enter a name.	The system request information in form: - Class Name : input name class.
2	Class admins, super admins select Create	The system returns information: Search training program : user input data and search as auto-completed search. Search result will be filtered & displayed as inputted letters Time : choose start time and end time.

		Admin : choose the information drop down available. Fsu : +Choose the name available. +Choose email available. Attendee: + User choose Attendee type of class including Intern / Fresher / Online fee-fresher / Offline fee-fresher. + User to input planned attendee number. + User to input accepted attendee number. + User to input actual attendee number.
3	Class admins, super admins fill data and select save as draft button	System validates the information and is successful.

Alternative: N/A

Exceptions:

No	Cause	System Response
1	"Name class" is empty	Focus in the "Name class" field. Show class name is required
2	"Time" is empty	Focus in the "Time" field. Show required field
3	"Admin " is empty	Focus in the "Admin" field. Show required field

4	“Fsu ” is empty	Focus in the “Fsu '' field. Show required field
5	“Attendee type ” is empty	Focus in the “Attendee type” field. Show required field
6	“Planned attendee ” is empty	Focus in the “Planned attendee” field. Show required field
7	“Accepted attendee ” is empty	Focus in the “Accepted attendee” field. Show required field
8	“Actual attendee ” is empty	Focus in the “Actual attendee” field. Show required field

Relationships: N/A

Business Rules: N/A

2.2.3.4. <Super Admin, Class Admin> Delete syllabus (UC_54)

Use Case Diagram

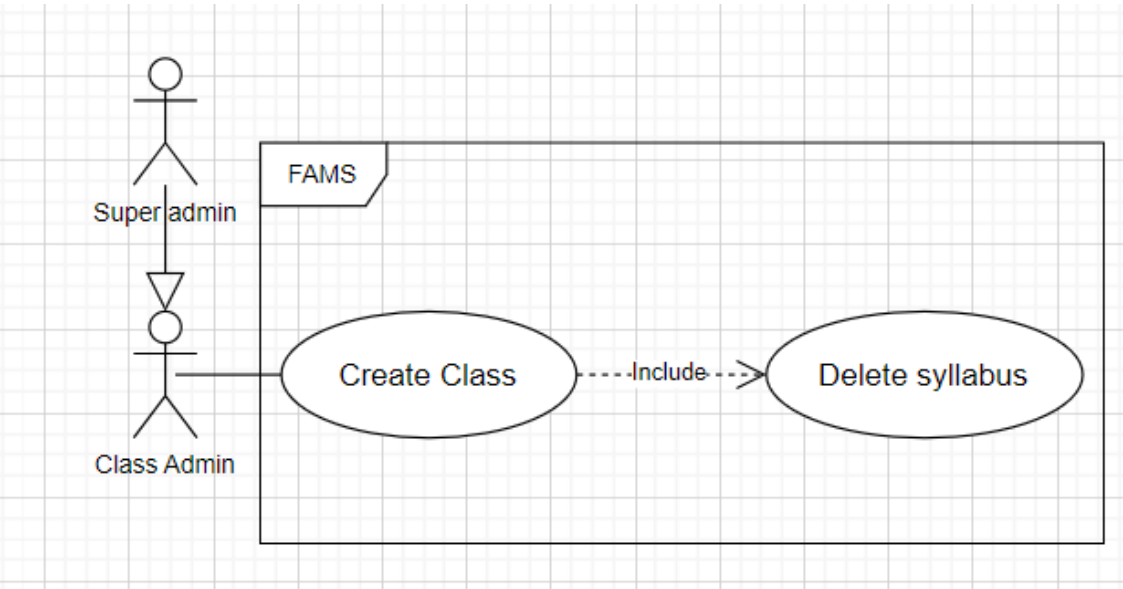


Figure 61- <Super Admin, Class Admin> Delete syllabus Use Case

Use Case Specification

USE CASE – UC_54

Use Case No.	54	**Use Case Version**	2.0
Use Case Name	Delete syllabus		
Author	Mai Văn Quốc Tịnh		
Date	18/03/2024	**Priority**	Normal

Actor:

- Class Admin, Super Admin.

Summary:

- This use case allows the admin, super admin to delete or add syllabus.

Goal:

- The account will be updated with new information

Triggers:

- Admin click on the "next" button after completing step 2 Main Success Scenario of USE CASE – UC_04 ".

Preconditions:

- Admin at the "Edit syllabus" page.

Post Conditions:

- Success: Users go to the Login page with their account and password have been registered.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Class admins, super admins click on the "next" button in create class.	The system requests information in the form of list syllabus.
2	Class admins, super admins click on the delete button	The system request "Do you really want to delete "Name syllabus" This class cannot be restored."
3	Class admins, super admins click on the delete button in pop up	The system request a list of syllabus information but there are no newly deleted

		syllabus
4	Class admins, super admins click on the "save" button	System validates the information and is successful.

Alternative:

AC 1: select "..." in class detail and choose edit button

+ AC 2: select edit button of training program.
+ AC 3: same as step 1 Main Success Scenario of USE CASE – UC_05.

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.3.5. <Super Admin, Class Admin> Edit Class (UC_55)

Use Case Diagram

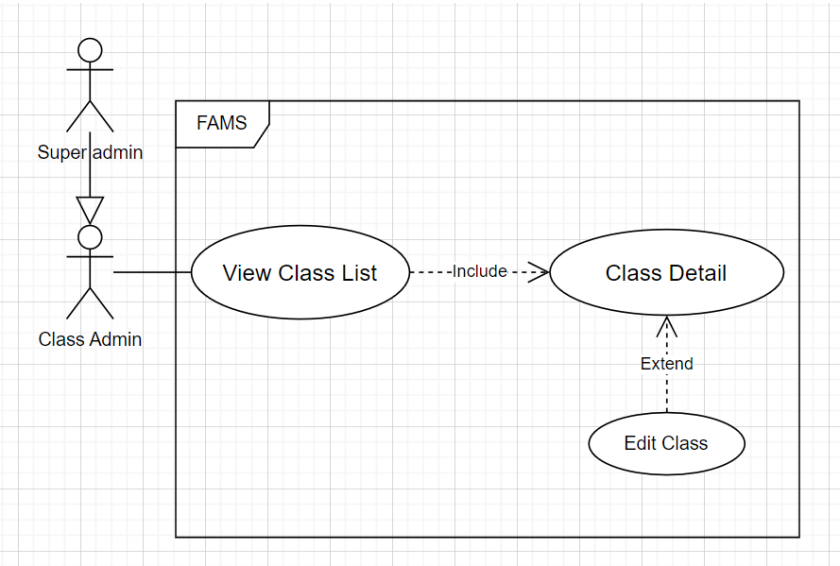


Figure 62 - <Super Admin, Class Admin> Edit Class Use Case

Use Case Specification

USE CASE – UC_55			
Use Case No.	55	**Use Case Version**	2.0
Use Case Name	Edit Class		
Author	Lê Việt Hùng		
Date	18/03/2024	**Priority**	Normal

Actor:

- Class Admin, Super Admin.

Summary:

- This use case allows the admin, super admin to edit class

Goal:

- The class will be updated with new information

Triggers:

- User click on the "..." button in List of class pages or Create Class Page and select "Edit User".

Preconditions:

- Have the role Admin or Super Admin (get role after login successful).

Post Conditions:

POST-1. Class adminor Super Admin has a been out to page and go to class list page

POST-2. New/edited info class will stored in system

POST-3. Remaining delivery capacity for the requested time window is updated.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Edit class" button.	The system request information in form: - Select Training Program: Choose 1 of list options - After select training program, system will display list of syllabus by training program
2	User fill out all the field	System will show the button to save or next page.
3	User click Save with training program and syllabus at now or can modify syllabus with the next button (Alternative 3.1)	The system will call service of backend and save info to database

Alternative: N/A

3.1 User can change Training Program

System will show a list of Training Program to pick

User choose program and press save

If success the system will display the program now.

User return to create class page

Info will store in database

Exceptions:

No	Cause	System Response
1	List Syllabus or Program is empty	Focus in "Syllabus" or "Program" field. Show error message

Relationships: N/A

Business Rules:

- Name: Must be format string.
- Hours and Day must be number

2.2.3.6. <Super Admin, Class Admin, Trainer> Search Calendar by week (UC_56)

Use Case Diagram

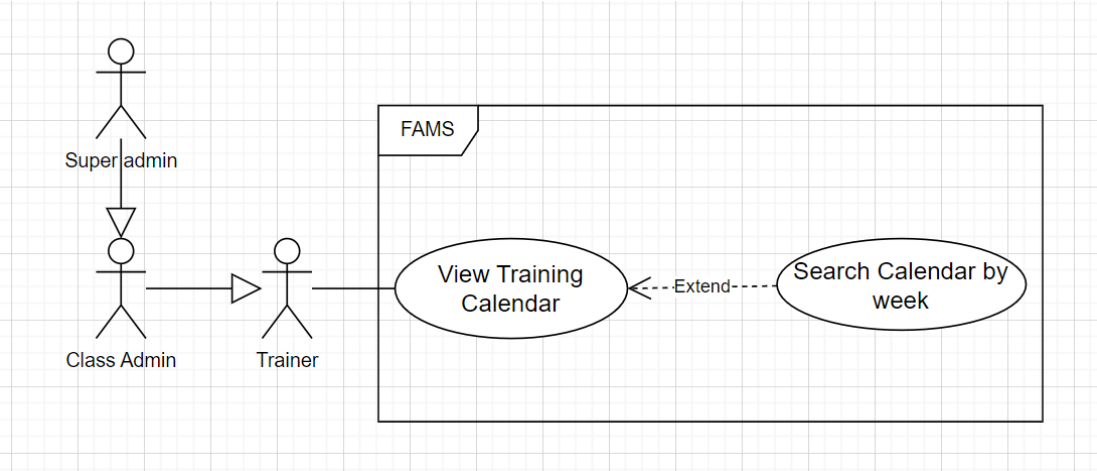


Figure 63 - <Super Admin, Class Admin, Trainer> Search Calendar by week Use Case

Use Case Specification

USE CASE – UC_56			
Use Case No.	56	**Use Case Version**	2.0
Use Case Name	Search Calendar by week		

Author	Lê Việt Hùng		
Date	18/03/2024	**Priority**	Normal

Actor:

- Class Admin, Super Admin.

Summary:

- This use case allows the admin or super admin to search calendar class in table calendar.

Goal:

- The class will be display item filtered

Triggers:

-

Preconditions:

- The user must have the role of Admin or Super Admin or Trainer (acquired after successful login).
- There must be at least one class available in the system.

Post Conditions:

- POST-1. The system displays events occurring within the specified week in the calendar..

Main Success Scenario:

Step	Actor Action	System Response
1	User selects the option to search by week in the calendar interface.	The system prompts the user to specify the desired week.
2	User specifies the week for the search (e.g., by providing a date within the week).	The system marks the class as deactivated.
3	The system updates the calendar view to display events occurring within the specified week.	The user can view events scheduled for the specified week in the calendar.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Events must be scheduled in the calendar for the search to be meaningful.
- The user must provide valid input for specifying the week.

2.2.3.7. <Super Admin, Class Admin, Trainer> Filter Calendar by day (UC_57)

Use Case Diagram

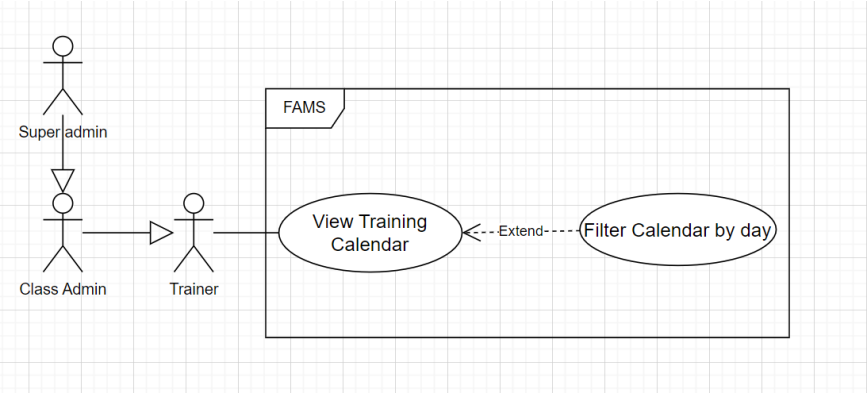


Figure 64 - <Super Admin, Class Admin, Trainer> Filter Calendar by day Use Case

Use Case Specification

USE CASE – UC_57			
Use Case No.	57	**Use Case Version**	2.0
Use Case Name	Filter Calendar by day		
Author	Lê Việt Hùng		
Date	18/03/2024	**Priority**	Normal

Actor:

- Class Admin, Super Admin

Summary:

- This use case allows the admin or super admin to filter classes based on certain criteria..

Goal:

- To display classes based on the selected filter criteria.

Triggers:

.- User clicks on the filter button in the List of Class pages.

Preconditions:

- The user must have the role of Admin or Super Admin or Trainer (acquired after successful login). There must be at least one class available in the system.

Post Conditions:

- POST-1. The list of classes is updated according to the selected filter criteria

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on the filter button.	The system displays filter options by day or hours or attendant type or location.
2	User selects filter criteria (e.g., active, inactive, by date, by day or hours or attendant type or location).	The system filters the list of classes accordingly.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Filter criteria must be provided.
- Classes must exist in the system to be filtered.

2.2.3.8. <Super Admin, Class Admin>Delete Class (UC_58)

Use Case Diagram

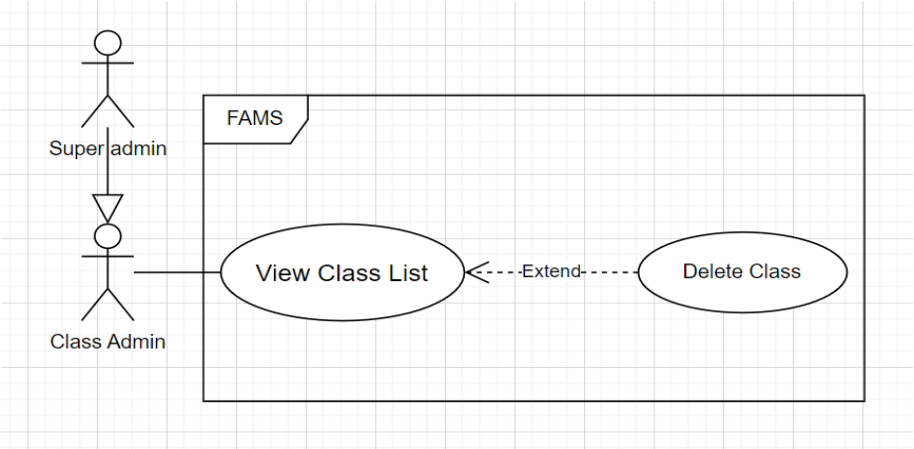


Figure 65 - <Super Admin, Class Admin> Delete Class Use Case

Use Case Specification

USE CASE – UC_58			
Use Case No.	58	**Use Case Version**	2.0
Use Case Name	Delete Class		
Author	Lê Việt Hùng		
Date	18/03/2024	**Priority**	Normal

Actor:

- Class Admin, Super Admin.

Summary:

- This use case allows the admin or super admin to delete a class from the system..

Goal:

- To permanently remove a class from the system.

Triggers:

- User selects the option to delete a class from the List of Class pages..

Preconditions:

- The user must have the role of Admin or Super Admin (acquired after successful login).
- The class must exist in the system.

Post Conditions:

POST-1. The class is permanently removed from the system.

Main Success Scenario:

Step	Actor Action	System Response

1	User selects the class to delete.	The system prompts for confirmation.
2	User confirms the deletion action.	The system permanently removes the class from the system.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Class must exist in the system to be deleted.

Deletion action is irreversible.

2.2.3.9. <Super Admin, Class Admin, Trainer> Class Detail in Calendar (UC_59)

Use Case Diagram

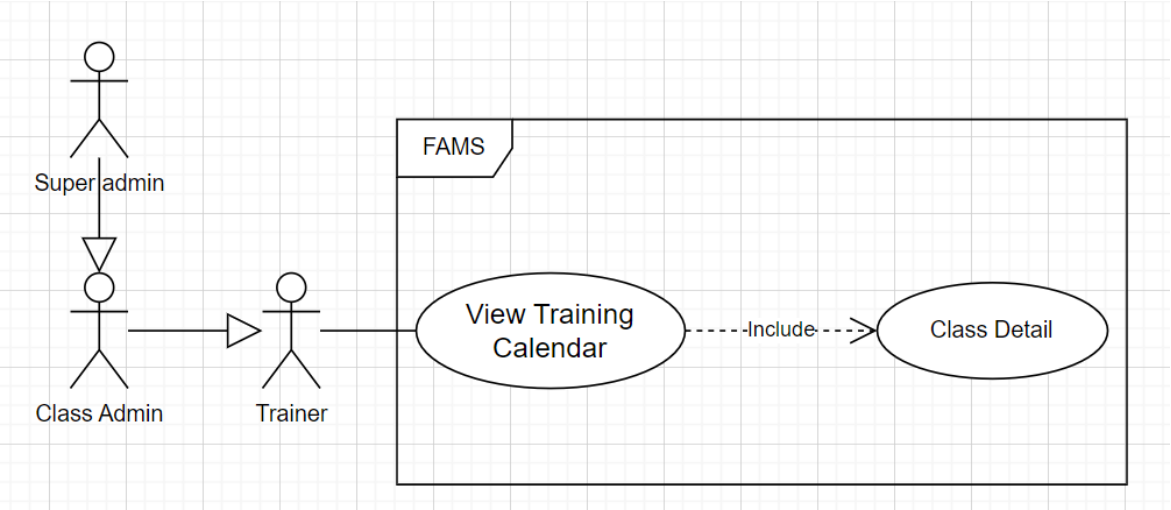


Figure 66 - <Super Admin, Class Admin, Trainer> Class Detail in Calendar Use Case

Use Case Specification

USE CASE – UC_59			
Use Case No.	59	**Use Case Version**	2.0
Use Case Name	Class Detail in Calendar		
Author	Lê Việt Hùng		
Date	18/03/2024	**Priority**	Normal

Actor:

- Class Admin, Super Admin.

Summary:

- This use case allows a user to view a summary of a class within the calendar page..

Goal:

- To provide users with essential information about a class directly from the calendar view.

Triggers:

- User hovers over a class event in the calendar..

Preconditions:

The user has access to the calendar page.

The class event must be displayed on the calendar.

Post Conditions:

POST-1. The system displays the summary of the selected class.

Main Success Scenario:

Step	Actor Action	System Response
1	User hovers over a class event in the calendar.	The system retrieves the summary of the selected class.
2	The system displays a summary of the class (e.g., class name, date, time, location, admin, trainer) in a tool tip or popup.	The user can view the summary of the selected class.

Alternative: N/A

Exceptions:

Relationships: N/A

Business Rules:

- Class event must be displayed on the calendar for viewing.
- Summary information of the class must be available in the system.

2.2.3.10. <Super Admin, Class Admin, Trainer> Filter Class List (UC_60)

Use Case Diagram

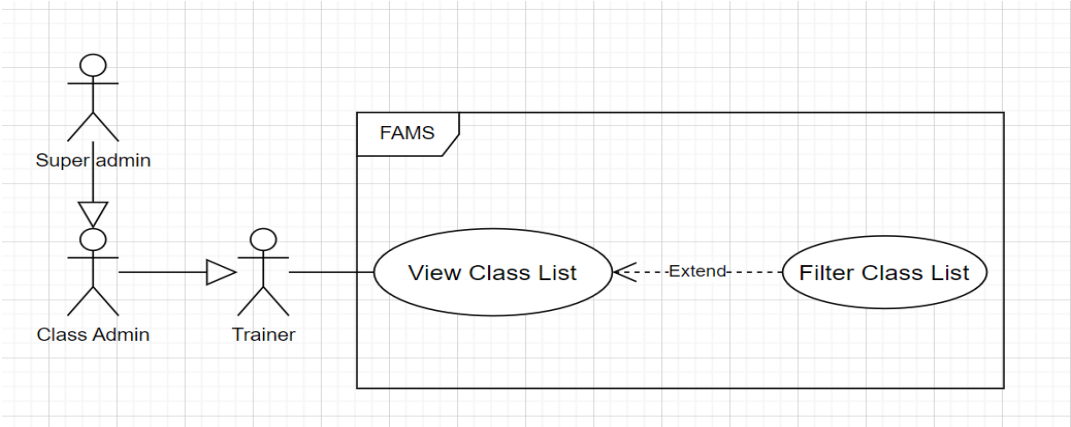


Figure 67- <Super Admin, Class Admin, Trainer> Filter Class List Use Case

Use Case Specification

USE CASE – UC_60			
Use Case No.	60	**Use Case Version**	2.0
Use Case Name	Filter Class List		
Author	Lê Việt Hùng		
Date	18/03/2024	**Priority**	Normal

Actor:

- Class Admin, Super Admin, Trainer

Summary:

- This use case allows the admin or super admin to filter the list of classes based on specific criteria.

Goal:

- To provide users with the ability to narrow down the list of classes based on their preferences..

Triggers:

- User selects the option to filter the class list from the class management interface..

Preconditions:

- The user must have the role of Admin or Super Admin or Trainer (acquired after successful login).

There must be at least one class available in the system.

Post Conditions:

POST-1. The system displays a filtered list of classes based on the selected criteria. Main Success Scenario:

Main Success Scenario:

Step	Actor Action	System Response
1	User selects the option to filter the class list.	The system displays filter options (e.g., by date, by program, by instructor).
2	User selects the desired filter criteria.	The system applies the selected filter criteria to the class list.
3	The system updates the class list to display only the classes that match the selected criteria.	The user can view the filtered list of classes.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Filter criteria must be provided.
- The class list must be updated accordingly based on the selected filter criteria.

2.2.3.11. <Super Admin, Class Admin> Duplicated Class (UC_60)

Use Case Diagram

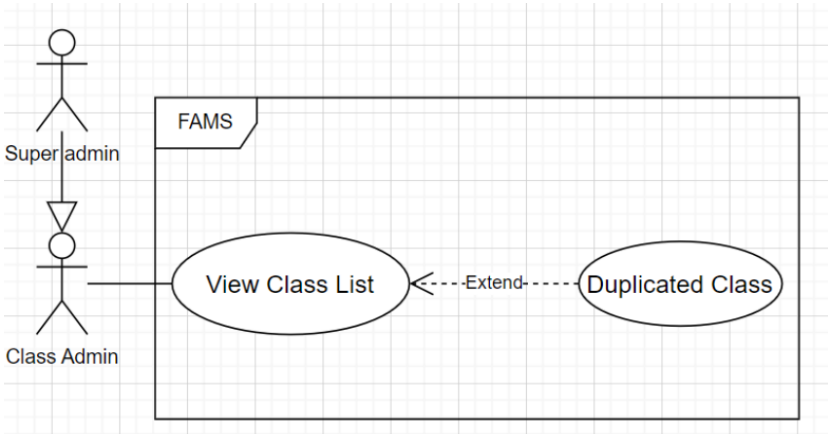


Figure 68 - <Super Admin, Class Admin> Duplicated Class Use Case

Use Case Specification

USE CASE – UC_60			
Use Case No.	60	**Use Case Version**	2.0
Use Case Name	Duplicated Class		
Author	Lê Việt Hùng		
Date	18/03/2024	**Priority**	Normal

Actor:

- Class Admin, Super Admin

Summary:

- This use case allows the admin or super admin to duplicate a class in the system.

Goal:

- To create a copy of an existing class with similar details.

Triggers:

- User selects the option to duplicate a class from the class management interface.

Preconditions:

The user must have the role of Admin or Super Admin (acquired after successful login).
The class to be duplicated must exist in the system...

Post Conditions:

The user must have the role of Admin or Super Admin (acquired after successful login).
The class to be duplicated must exist in the system.

Main Success Scenario:

Step	Actor Action	System Response
1	User selects the class to duplicate.	The system retrieves the details of the selected class.
2	User initiates the duplication process..	User initiates the duplication process.
3	User reviews the duplicate class in class page	The system create new class in display it in class list and reload list class.

Alternative: N/A

Exceptions:

Relationships: N/A

Business Rules:

- Class must exist in the system to be duplicated.
- The duplicate class should have similar details as the original class.
- Duplicate class creation should be successful without errors.

2.2.3.12. <Super Admin, Class Admin, Trainer> Search on Training Calendar (UC_61)

Use Case Diagram

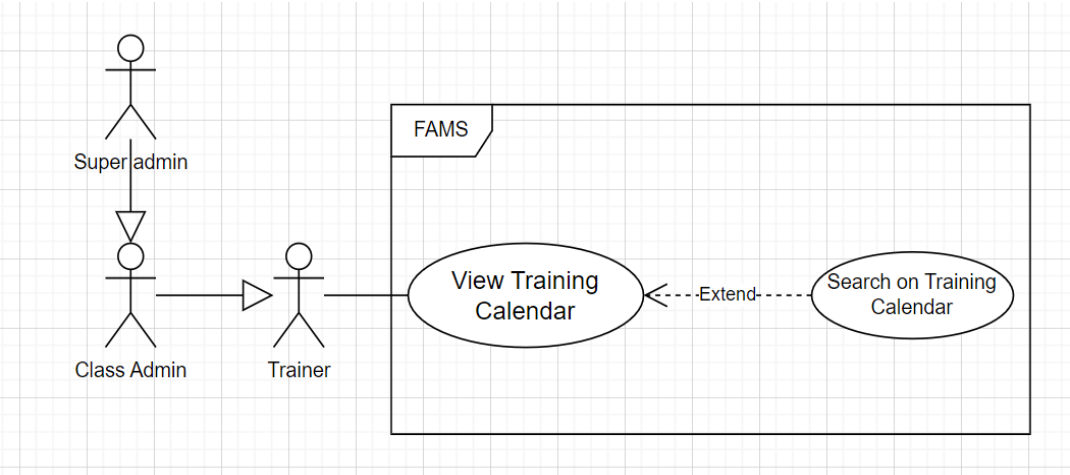


Figure 69 - <Super Admin, Class Admin, Trainer> Search on Training Calendar Use Case

Use Case Specification

USE CASE – UC_61			
Use Case No.	61	**Use Case Version**	2.0
Use Case Name	Search on Training Calendar		
Author	Nguyễn Công Thành		
Date	19/03/2024	**Priority**	High

Actor:

- Class Admin, Super Admin, Trainer.

Summary:

- This use case allows the admin, super admin, trainer to search on the training calendar to quickly find the information they need on the schedule.

Goal:

- To allow the admin to effectively search and retrieve specific information from the training calendar within the system

Triggers:

- Admin enter a search query in the search bar

Preconditions:

- Admin is logged in and is at the "Training Calendar" page

Post Conditions:

- Success: Admin is presented with search results relevant to their query

- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin enters a search query in the training calendar search bar	The system retrieves and displays search results relevant to the query
2	Admin views the search query	System presents the search results clearly the calendar

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Class schedules must exist in the system to be viewed.
- Admin must be logged in to view the training calendar.

2.2.3.13. <Super Admin, Class Admin, Trainer> View Training Calendar (UC_62)

Use Case Diagram

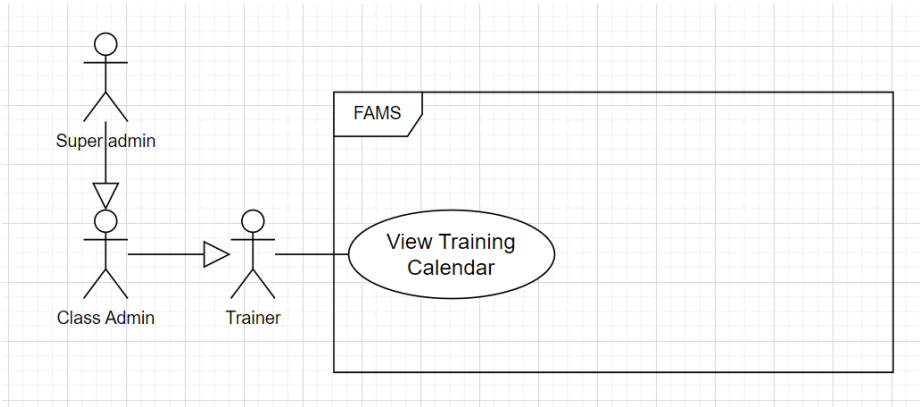


Figure 70 - <Super Admin, Class Admin, Trainer> View Training Calendar Use Case

Use Case Specification

USE CASE – UC_62			
Use Case No.	62	**Use Case Version**	2.0
Use Case Name	View Training Calendar		
Author	Nguyễn Công Thành		
Date	19/03/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin, Trainer

Summary:

- This use case allows the Admin to view the schedule of each class via the training **calendar.**

Goal:

- To provide an overview of the schedule for a specific class.

Triggers:

- Admin navigates to the training calendar page.

Preconditions:

- Admin is logged in and is at the "Training Calendar" page
- The class schedules must exist in the system.

Post Conditions:

- Success: Admin is presented with search results relevant to their query
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin enters a search query in the training calendar search bar	The system displays the training calendar with the schedules of all classes.
2	Admin selects a specific class	The system highlights the schedule for the selected class.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Class schedules must exist in the system to be viewed.
- Admin must be logged in to view the training calendar.

2.2.3.14. <Super Admin, Class Admin, Trainer> Sort Class (UC_63)

Use Case Diagram

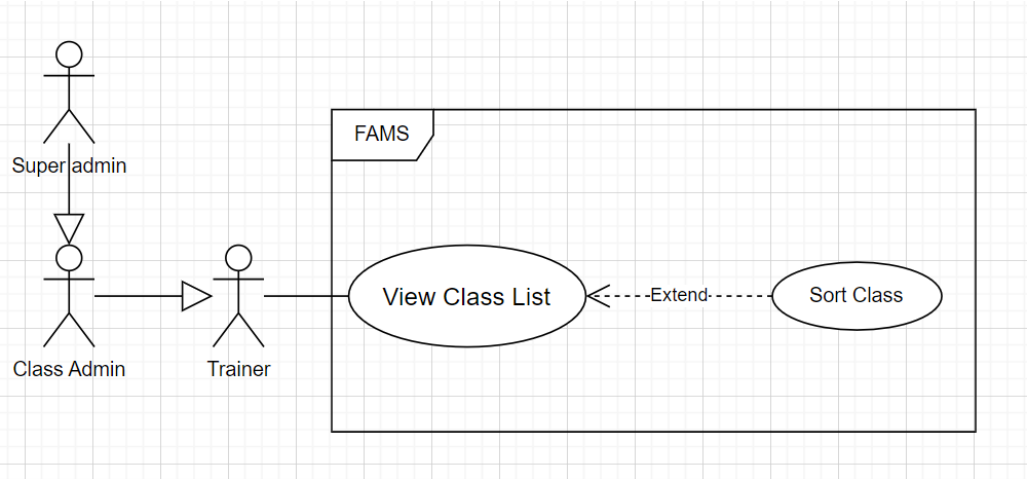


Figure 71 - <Super Admin, Class Admin, Trainer> Sort Class Use Case

Use Case Specification

USE CASE – UC_63			
Use Case No.	63	**Use Case Version**	2.0
Use Case Name	Sort Class		
Author	Lê Việt Hùng		
Date	19/03/2024	**Priority**	High

Actor:

- Super Admin, Class Admin, Trainer

Summary:

- This use case allows the admin or super admin to sort the list of classes based on multiple criteria.

Goal:

- To provide users with the ability to organize the class list according to their preferences.

Triggers:

- User selects the option to sort the class list from the class management interface..

Preconditions:

- Admin is logged in and is at the "Class list" page
- The class schedules must exist in the system.

Post Conditions:

POST-1. The system displays a sorted list of classes based on the selected sorting criteria.

Main Success Scenario:

Step	Actor Action	System Response
1	User selects the option to sort the class list.	The system displays the The system displays sorting options (e.g., by name, by date, by instructor).schedule of the selected class.
2	User selects the desired sorting criteria (e.g., sort by name, then by date).	The system applies the selected sorting criteria to the class list.
3	The user can view the sorted list of classes.	The system updates the class list to display classes sorted according to the selected criteria.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Sorting criteria must be provided.
- The class list must be updated accordingly based on the selected sorting criteria.

2.2.3.15. <Super Admin, Class Admin>Elasticsearch (UC_64)

Use Case Diagram

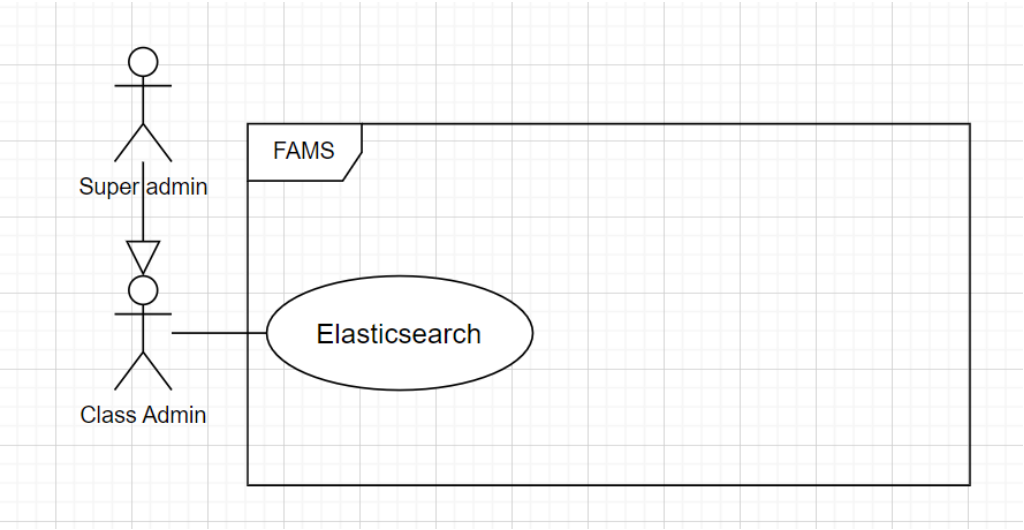


Figure 72 - <Super Admin, Class Admin> Elasticsearch Use Case

Use Case Specification

USE CASE – UC_64			
Use Case No.	64	**Use Case Version**	2.0
Use Case Name	Elasticsearch		
Author	Lê Việt Hùng		
Date	19/03/2024	Priority	High
Actor:			

- Super Admin, Class Admin

Summary:

- This use case involves implementing Elasticsearch to create a product search engine for an e-commerce website..

Goal:

- To provide users with fast and relevant search results when searching for products on the website.

Triggers:

- User initiates a search for products on the website.

Preconditions:

- Elasticsearch is installed and configured.
- List Class data is indexed in Elasticsearch.

Post Conditions:

- POST-1. Elasticsearch returns relevant search results to the user.

Main Success Scenario:

Step	Actor Action	System Response
1	User enters a search query in the search bar on the website.	The system sends the search query to Elasticsearch.
2	Elasticsearch processes the search query and retrieves relevant product data from its index.	Elasticsearch returns the search results to the system.
3	The system displays the search results to the user.	The user can view relevant products matching their search query.

Alternative: N/A

Exceptions:

- Elasticsearch does not find any products matching the search query.
- The system displays a message to the user indicating no relevant products were found.

Relationships: N/A

Business Rules:

- Product data must be properly indexed in Elasticsearch.
- Elasticsearch queries must be optimized for performance and relevance.
- The search engine must handle a large volume of concurrent search queries efficiently.

2.2.3.16. <Super Admin, Class Admin, Trainer> Export Class List (UC_65)

Use Case Diagram

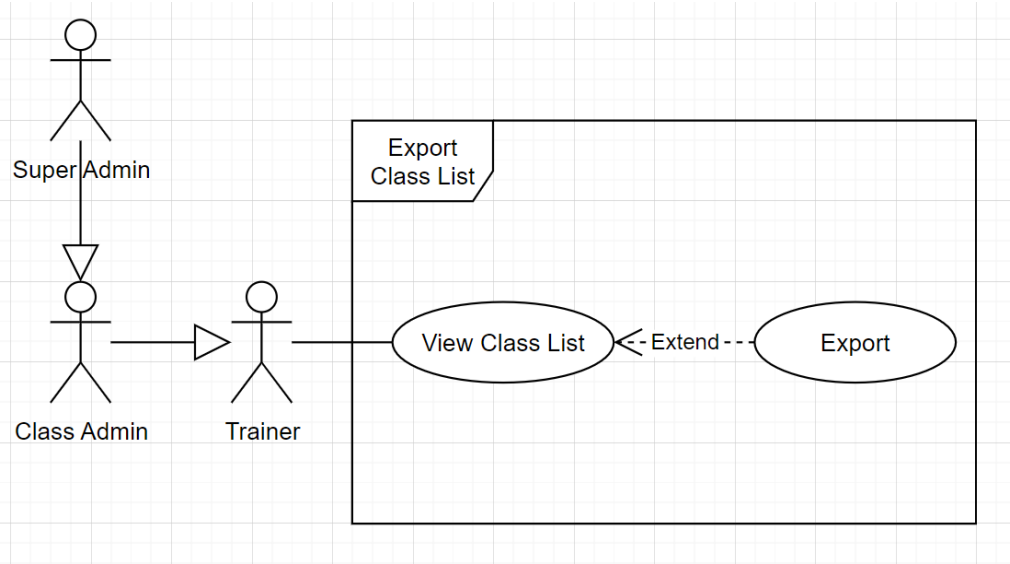


Figure 73 - <Super Admin, Class Admin, Trainer> Export Class List

Use Case Specification

USE CASE – UC_65			
Use Case No.	65	**Use Case Version**	2.0
Use Case Name	Export		
Author	Mai Văn Quốc Tịnh		
Date	15/04/2024	**Priority**	High

Actor:

- Super Admin, Class Admin, Trainer

Summary:

- This use case allows the user to export all classes.

Goal:

- Users can export all training programs as csv files.

Triggers:

- User clicks on "Export" label to download class list.

Preconditions:

- User must login with 3 role "Super Admin, Class Admin, Trainer"
- System displays a success notification.

Post Conditions:

- **Success:** Download successfully all training program list in csv file
- **Fail:** Can't export file

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Export" button	• The system validates data and download to local storage •

Alternative: N/A
Exceptions:N/A
Relationships: N/A
Business Rules: N/A

2.2.5. Student Information Management Over View Use Case

Use Case Diagram

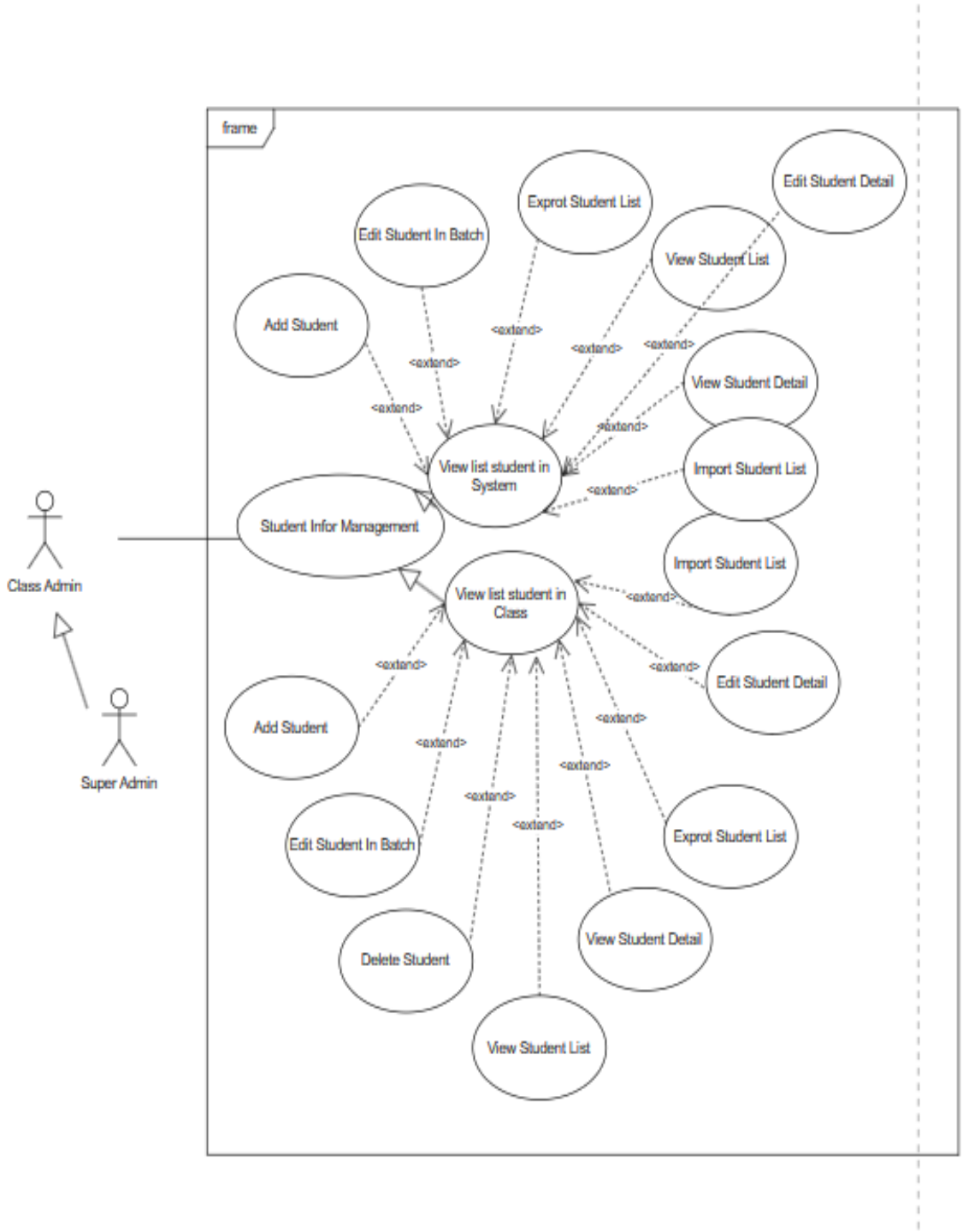


Figure 74 - Student Information Management Over View Use Case

2.2.5.1. <Super Admin, Class Admin>View list students in System (UC_66)

Use Case Diagram

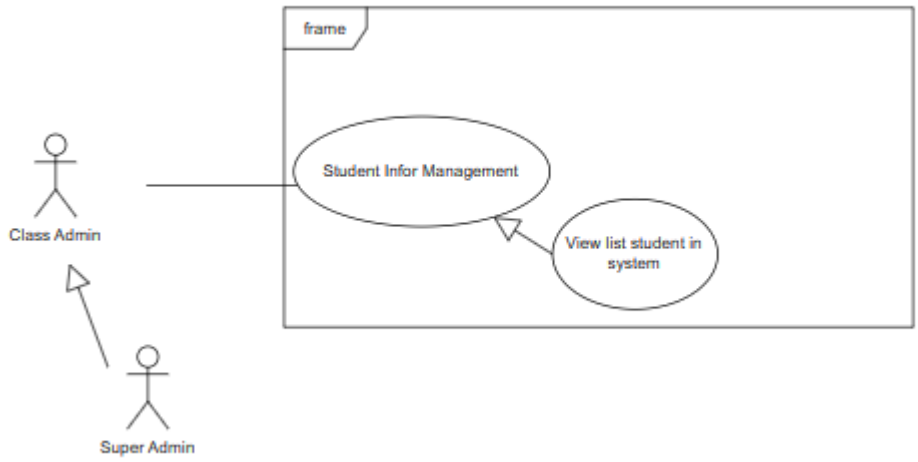


Figure 75 - <Super Admin, Class Admin > View list of students by system

Use Case Specification

USE CASE – UC_66			
Use Case No.	66	**Use Case Version**	2.0
Use Case Name	View Student List In System		
Author	Nguyễn Đỗ Chung Quý		
Date	20/3/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin

Summary:

- This use case allows Super Admin and Class Admin to view list of students in a system.

Goal:

- Get information of all students in the specific system.

Triggers:

- User click on “Student List” button.

Preconditions:

- User at "Class Detail Screen"

Post Conditions:

- **Success:** Display list student in system.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Student List" button.	The system display the list of students in that class.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.5.2. <Super Admin, Class Admin>View list students in class (UC_67)

Use Case Diagram

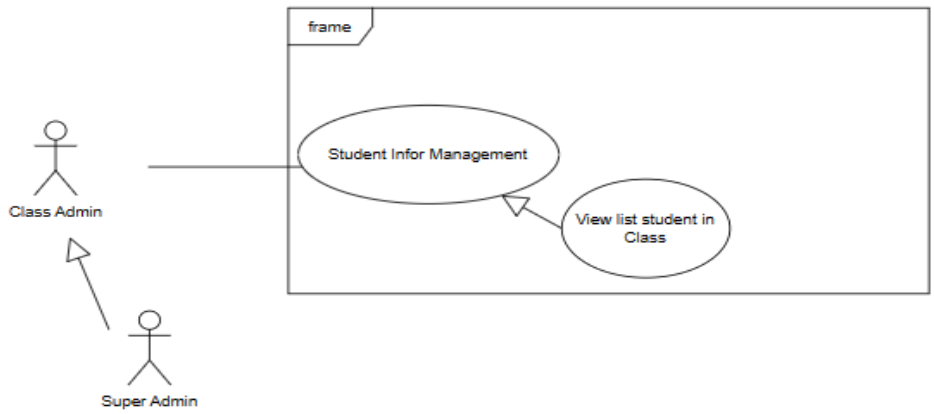


Figure 76 - <Super Admin, Class Admin > View list of students by class

Use Case Specification

USE CASE – UC_67			
Use Case No.	67	**Use Case Version**	2.0
Use Case Name	View Student List In Class		

Author	Nguyễn Sơn Lam		
Date	20/3/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin

Summary:

- This use case allows Super Admin and Class Admin to view list of students in a class.

Goal:

- Get information of all students in the specific class.

Triggers:

- User click on "Student List" button.

Preconditions:

- User at "Class Detail Screen"

Post Conditions:

- **Success:** Display list student in class.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Student List" button.	The system display the list of students in that class.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.5.3. <Super Admin, Class Admin>Edit the information and status of each student from student list in class (UC_68)

Use Case Diagram

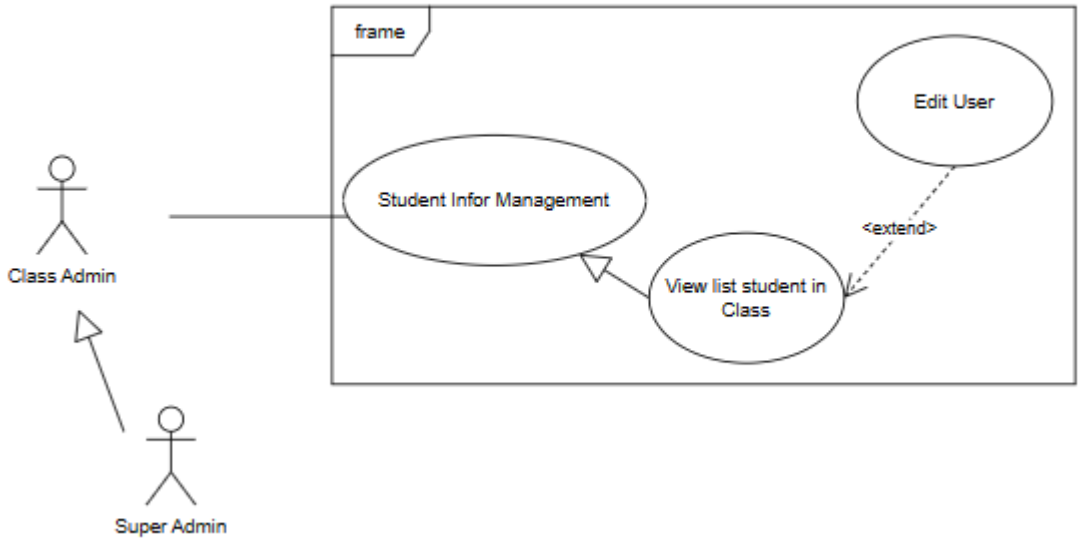


Figure 77 - <Super Admin, Class Admin> Edit the information and status of each student (access from student list in class detail)

Use Case Specification

USE CASE – UC_68			
Use Case No.	68	**Use Case Version**	2.0
Use Case Name	Edit Student Detail In Class		
Author	Nguyễn Sơn Lam		
Date	20/3/2024	**Priority**	Normal
Actor: ● Super Admin, Class Admin. **Summary:** ● Allow User to update the information/status of each student. **Goal:** ● Updating the newest and exact information/status of each student. **Triggers:**			

- User click on “Edit student” button.

Preconditions:

- User at “Student List In A Class Screen”

Post Conditions:

- **Success:** User edit student’s information successfully.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on “Student List” button.	The system display the general student’s information and information of student in that class.
2	User click on “ ” icon to turn on edit mode.	The system change the text to form input for admin to update information.
3	User click on “Save” icon to update the newest student’s information.	The system display update success message.

Alternative: N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1	“Name” is empty	Focus in “Name” field. Show error message
2	“Phone” is empty	Focus in “Phone” field. Show error message
3	“Email” is empty	Focus in “Email” field. Show error message
4	“RECer” is empty	Focus “RECer” field. Show error message

[Exceptions 2]

No	Cause	System Response
1	Email is not contain @.	Show error message.
2	The first number of phone number is different with 0.	Show error message.
3	Phone number length different with 10.	Show error mesage

Relationships: N/A

Business Rules:

- Email: Contain "@".
- Phone number must start with 0.
- Phone number length must be equal 10

2.2.5.4. <Super Admin, Class Admin>View Each Student (UC_69)

Use Case Diagram

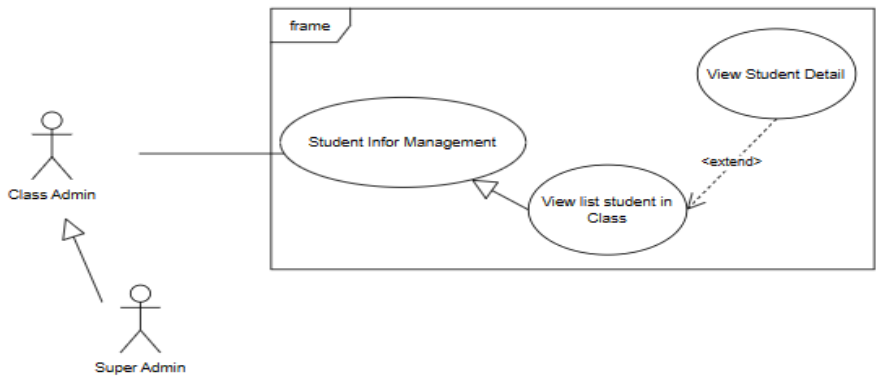


Figure 78 - <Super Admin, Class Admin> View each student detail

Use Case Specification

USE CASE – UC_69

Use Case No.	69	**Use Case Version**	2.0
Use Case Name	view each student		
Author	Nguyễn Sơn Lam		
Date	20/3/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin.

Summary:

- Allow user to see each student.

Goal:

- User view each student.

Triggers:

- User click on "Student List" button.

Preconditions: N/A

Post Conditions: N/A

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Student List" button.	The system display the general student's information and information of student in that class.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.5.5. <Super Admin, Class Admin>Import Data (UC_70)

Use Case Diagram

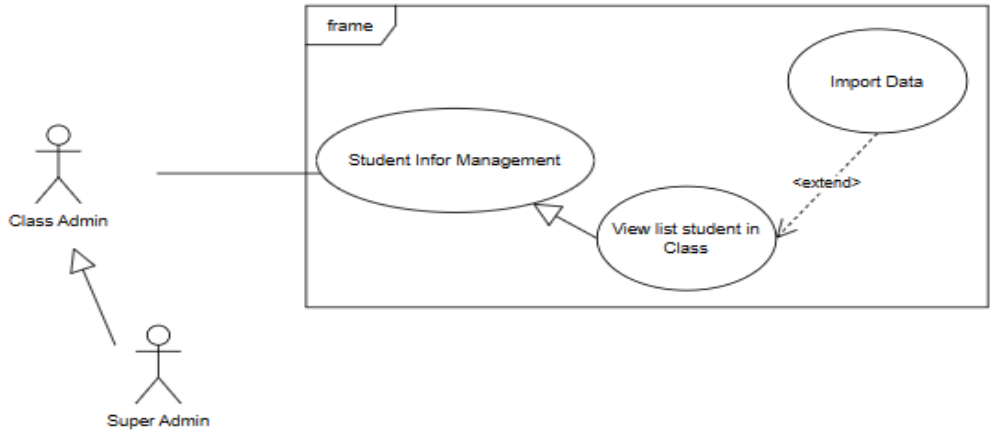


Figure 79 - <Super Admin, Class Admin> Import Data

Use Case Specification

USE CASE – UC_70			
Use Case No.	70	**Use Case Version**	2.0
Use Case Name	Import data		
Author	Nguyễn Lê Viên Minh		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin.

Summary:

- This use case allows User to import data Excel file into the system.

Goal:

- User import data from Excel file into the system. If there is a duplicate, the three options will be displayed to the User.

Triggers:

- User clicks on “Import” button.

Preconditions:

- User at “Import” popup.

Post Conditions:

- **Success:** User import succeeds if the system contains no data.

- **Fail:** The System will show the User the following error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	User click “Select” button.	The system requests file identification information from the “.xlsx” file extension.
2	User click “Download” link.	The system will give User one template file located in the same row as the search bar.
3	User click “Import” button.	The system will return “ File uploaded successfully!” if the system does not contain data and the imported file does not contain duplicate data.
4	User click “Cancel” button.	The popup will exit and return to the main screen.
5	User choose duplicate option “Allow”	The system will show the message “Data uploaded successfully with 'Allow' option.”.
6	User choose duplicate option “Replace”	The system will show the success message “Data uploaded successfully with 'Replace' option.”.
7	User choose duplicate option “Skip”	The system will shows the success message “Data uploaded successfully with 'Skip' option.”

Alternative: N/A

Exceptions:

No	Cause	System Response
1	User “does not select file”.	System shows error message “Please select a file.”
2	User “import data duplicate”.	System shows error message “Data has been duplicated.”
3	User “selects a file that	System shows error message “Error

	contains corrupted data or does not match the system".	uploading file. Please try again.".

Relationships: N/A

Business Rules:

- **File Format Compatibility:** The system only accepts Excel files with the ".xlsx" extension for import.

- **Error Handling:** The system must display appropriate error messages in case of various scenarios such as failure to select a file, duplicate data, or file corruption.

- **Data Duplication Handling:** If duplicate data is detected during import, the system provides the User with three options: "Allow", "Replace", or "Skip".

- **System Data Check:** The import process succeeds only if the system contains no existing data. If data already exists, the system requires the user to handle duplicate entries.

- **Success Confirmation:** Upon successful import, the system displays a confirmation message indicating the successful upload and chosen duplicate handling option.

2.2.5.6. <Super Admin, Class Admin>Edit Student from Student System List (UC_71)

Use Case Diagram

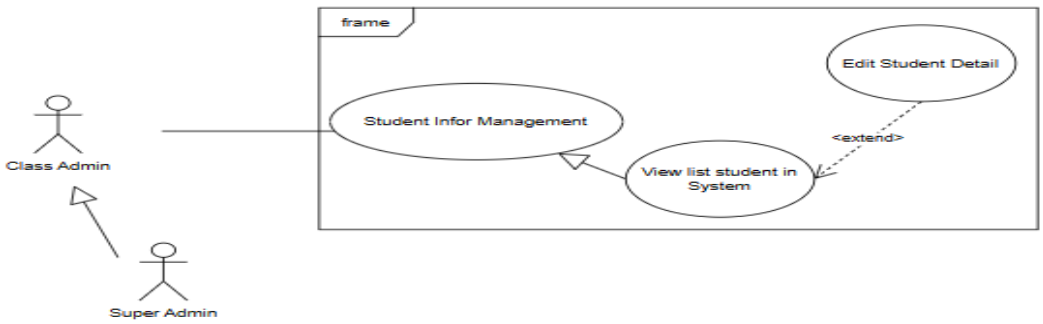


Figure 80 - <Super Admin, Class Admin > Edit Student Detail

Use Case Specification

USE CASE – UC_71			
Use Case No.	71	**Use Case Version**	2.0
Use Case Name	Edit Student System Detail		
Author	Nguyễn Minh Hiền		
Date	20 /3/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin.

Summary:

- Allow User to update the information of each student.

Goal:

- Updating the newest and exact information of each student.

Triggers:

- User click on "Edit student" button.

Preconditions:

- User at "Student List System Screen"

Post Conditions:

- **Success:** User edit student's information successfully.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Student List" button.	The system display the general list of all student in system .
2	User click on "⋯ " icon.	The system show out the list of 3 action contain edit student for admin to choose.
3	User click on " ✎ " .	The system pop-up a form text contain detail of that student and allow admin to change.
4	User click on "Save" icon to update the newest student's information.	The system display update success message.

Alternative: N/A

Exceptions:
[Exception 1]

No	Cause	System Response
1	"Name" is empty	Focus in "Name" field. Show error message
2	"Phone" is empty	Focus in "Phone" field. Show error message
3	"Email" is empty	Focus in "Email" field. Show error message
4	"RECer" is empty	Focus "RECer" field. Show error message

[Exceptions 2]

No	Cause	System Response
1	Email is not contain @.	Show error message.
2	The first number of phone number is different with 0.	Show error message.
3	Phone number length different with 10.	Show error mesage

Relationships: N/A

Business Rules:

- Email: Contain "@".
- Phone number must start with 0.
- Phone number length must be equal 10

2.2.5.7. <Super Admin, Class Admin>Edit the information and status of each student from student list in class (UC_72)

Use Case Diagram

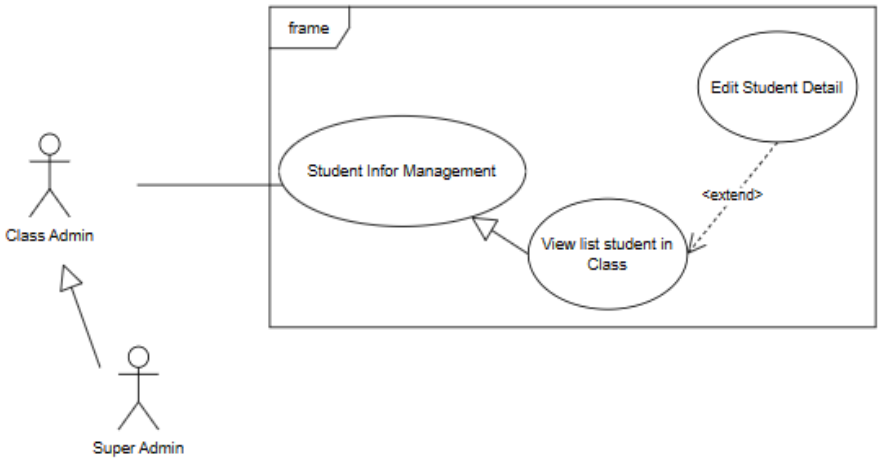


Figure 81 - <Super Admin, Class Admin> Edit the information and status of each student from student list in class detail

Use Case Specification

USE CASE – UC_72			
Use Case No.	72	**Use Case Version**	2.0
Use Case Name	Edit Student Status In Class		
Author	Nguyễn Sơn Lam		
Date	20/3/2024	**Priority**	Normal
Actor: ● Super Admin, Class Admin. **Summary:** ● Allow User to update the information/status of each student. **Goal:** ● Updating the newest and exact information/status of each student. **Triggers:** ● User click on “Edit student” button.			

Preconditions:

- User at "Student List In A Class Screen"

Post Conditions:

- **Success:** User edit student's information successfully.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Student List" button.	The system display the general student's information and information of student in that class.
2	User click on "" icon to turn on edit mode.	The system change the text to form input for admin to update information.
3	User click on "Save" icon to update the newest student's information.	The system display update success message.

Alternative: N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1	"Name" is empty	Focus in "Name" fie ld. Show error message
2	"Phone" is empty	Focus in "Phone" field. Show error message
3	"Email" is empty	Focus in "Email" field. Show error message
4	"RECer" is empty	Focus "RECer" field. Show error message

[Exceptions 2]

No	Cause	System Response
1	Email is not contain @.	Show error message.
2	The first number of phone number is different with 0.	Show error message.
3	Phone number length different with 10.	Show error mesage

Relationships: N/A

Business Rules:

- Email: Contain "@".
- Phone number must start with 0.
- Phone number length must be equal 10

2.2.5.8. <Super Admin, Class Admin>Edit Student Status in batch_Class (UC_73)

Use Case Diagram

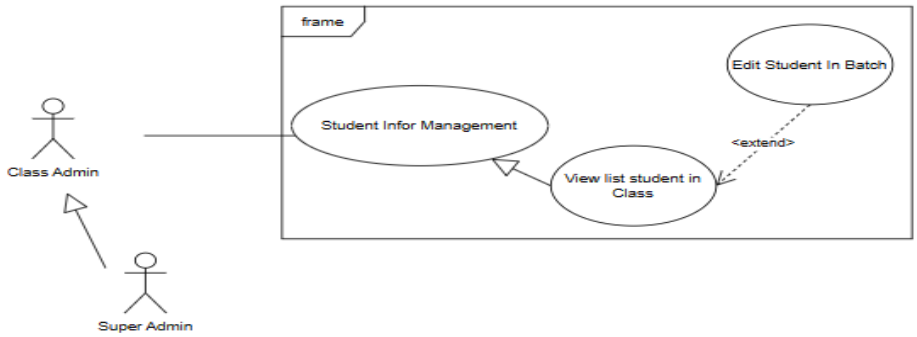


Figure 82 - <Super Admin, Class Admin> Edit Student Status in batch_class

Use Case Specification

USE CASE – UC_73			
Use Case No.	73	**Use Case Version**	2.0

Use Case Name	Edit Status of Students in Batch By Class		
Author	Nguyễn Minh Hiền		
Date	20/3/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin.

Summary:

- This use case allows the admin to update the attending status of multiple students.

Goal:

- Updating the status of multiple student.

Triggers:

- User click on “Edit student” button.

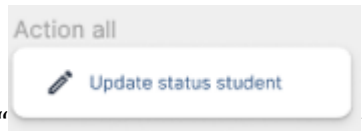
Preconditions:

- The attending status of selected students is updated.
- The class records the update date and the admin's account.

Post Conditions:

- **Success:** The class records the update date and the admin's account.
- **Fail:** An error message appears stating, "Please select students with the same attending status.".

Main Success Scenario:

Step	Actor Action	System Response
1	User click on “Student List” button.	The class screen display the general student’s information and information of student in that class.
2	User click on “ ☑ ” icon to choose student.	The admin selects multiple students from the list by checking checkboxes in the first column..
3	User click on “  ”	If all selected students have the same attending status: - A popup window appears showing changeable statuses

	button.	(Inactive, Off, Keep Class).
4	User clicks the "Save" button.	The class screen updates the attending status of the selected students.

Alternative:

1	If the admin tries to select students with different attending statuses and clicks the "Update status" button	The class screen deactivates the "Update status" button. A tooltip text appears when the admin hovers over the button, stating, "Can't update status student."

Exceptions:
[Exception 1]

No	Cause	System Response
1	select students with different attending statuses	The class screen deactivates the "Update status" button and provides a tooltip text when the admin hovers over the button, stating, "Can't update status student."

Relationships: N/A

Business Rules:

- Users can only update the attending status of multiple students if they share the same current attending status.
- The class screen should display only those status options that comply with the predefined rules for status transitions.

2.2.5.9. <Super Admin, Class Admin>Edit Student Status in batch_System (UC_74)

Use Case Diagram

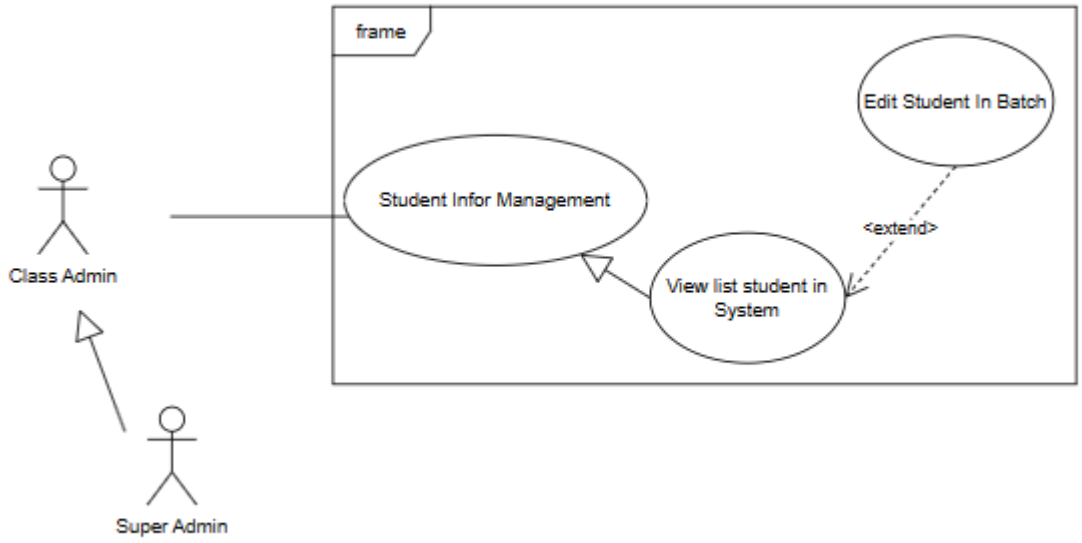


Figure 83 - <Super Admin, Class Admin> Edit Student Status in batch_system

Use Case Specification

USE CASE – UC_74			
Use Case No.	74	**Use Case Version**	2.0
Use Case Name	Edit Status of Students in Batch By System		
Author	Nguyễn Minh Hiền		
Date	20/3/2024	**Priority**	Normal

Actor:

- Admin.

Summary:

- This use case allows the admin to update the attending status of multiple students.

Goal:

- Updating the status of multiple student.

Triggers:

- User click on “Edit student” button.

Preconditions:

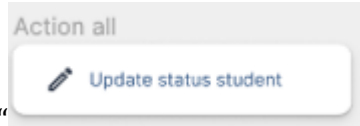
- The attending status of selected students is updated.

- The system records the update date and the admin's account.

Post Conditions:

- **Success:** The system records the update date and the admin's account.
- **Fail:** An error message appears stating, "Please select students with the same attending status.".

Main Success Scenario:

Step	Actor Action	System Response
1	User click on “Student List” button.	The system screen display the general student’s information and information of student in that class.
2	User click on “ ☑ ” icon to choose student.	The admin selects multiple students from the list by checking checkboxes in the first column..
3	User click on “  ” button.	If all selected students have the same attending status: - A popup window appears showing changeable statuses (Inactive, Off, Keep Class).
4	User clicks the "Save" button.	The system screen updates the attending status of the selected students.

Alternative:

1	If the User tries to select students with different attending statuses and clicks the "Update status" button	The system screen deactivates the "Update status" button. A tooltip text appears when the admin hovers over the button, stating, "Can't update status student."

Exceptions:

[Exception 1]

No	Cause	System Response
1	select students with different attending statuses	The system screen deactivates the "Update status" button and provides a

		tooltip text when the admin hovers over the button, stating, "Can't update status student."

Relationships: N/A

Business Rules:

- Users can only update the attending status of multiple students if they share the same current attending status.
- The system should display only those status options that comply with the predefined rules for status transitions.

2.2.5.10. <Super Admin, Class Admin>Export the information of each student from student list in system (UC_75)

Use Case Diagram

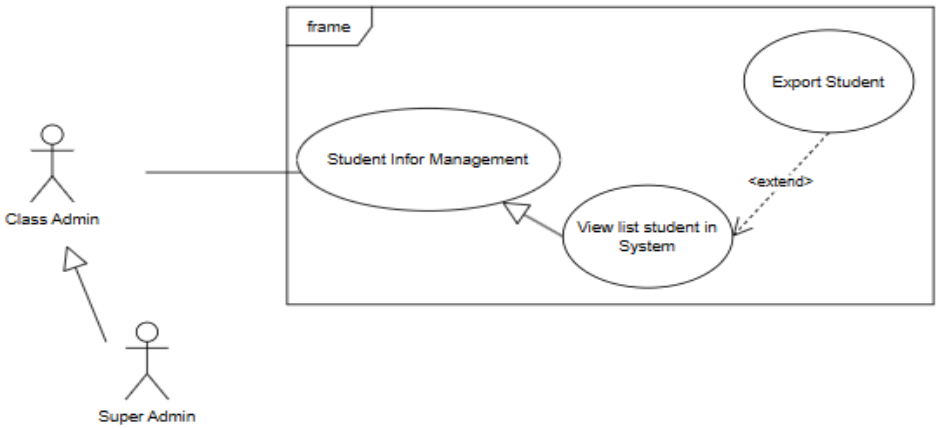


Figure 84 - <Class Admin, Super Admin>Export Data Excel In System

Use Case Specification

USE CASE – UC_75			
Use Case No.	75	**Use Case Version**	2.0
Use Case Name	Export Student List By System		
Author	Nguyen Manh Cuong		
Date	20/3/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin.

Summary:

- Allow user to export the information of each student.

Goal:

- Export exact information of each student in filter show on student system screen.

Triggers:

- User click on "Export" button.

Preconditions:

- User at "Student List In A Class Screen"

Main Success Scenario:

Step	Actor Action	System Response
1	User click on " Export " button.	File studentdata.xlxs will be dowload automatically

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.5.11. <Super Admin, Class Admin>Export the information of each student from student list in class (UC_76)

Use Case Diagram

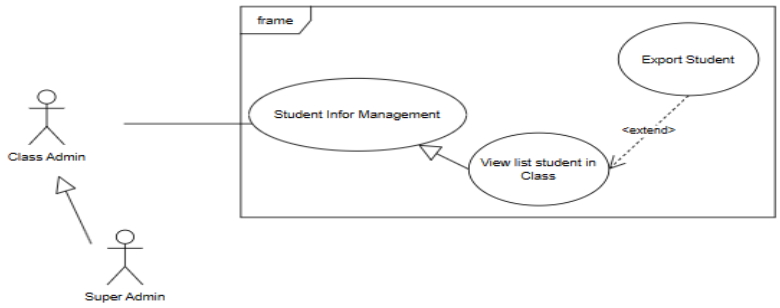


Figure 85 - <Class Admin, Super Admin> Export Data Excel

Use Case Specification

USE CASE – UC_76			
Use Case No.	76	**Use Case Version**	2.0
Use Case Name	Export Student List In Class		
Author	Nguyen Manh Cuong		
Date	20/3/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin.

Summary:

- Allow User to export the information of each student.

Goal:

- Export exact information of each student in filter show on student class screen.

Triggers:

- User click on "Export" button.

Preconditions:

- User at "Student List In A Class Screen"

Main Success Scenario:

Step	Actor Action	System Response
1	User click on " " button.	File studentdata.xlxs will be dowload automatically

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.5.12. <Super Admin, Class Admin>Delete of each student (UC_77)

Use Case Diagram

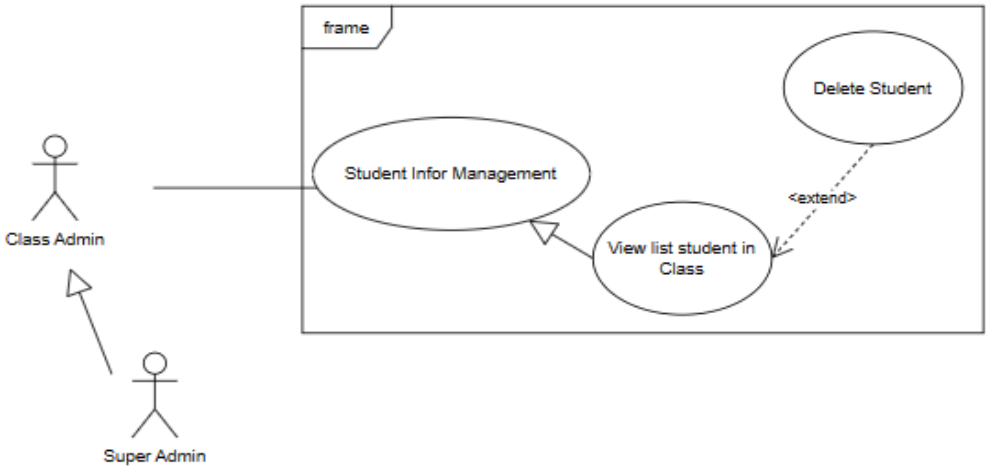


Figure 86 - <Class Admin, Super Admin>Delete Student

Use Case Specification

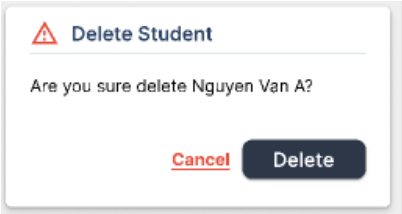
USE CASE – UC_77			
Use Case Name	Delete Student		
Use Case No.	77	**Use Case Version**	2.0
Author	Nguyễn Đỗ Chung Quý		
Date	20/3/2024	**Priority**	Normal
Actor: ● Super Admin, Class Admin. **Summary:** ● Allow User to delete the student. **Goal:** ● Delete exact each student. **Triggers:** ● User click on “Delete student” button. **Preconditions:**			

- User at "Student List In A Class Screen"

Post Conditions:

- **Success:** User remove student successfully.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Student List" button.	The system display the general student's information and information of student in that class.
2	User min click on " " icon.	Show the delete student dialog
3	User click on  delete	The student will be remove in class

Alternative: N/A

Exceptions:
[Exception 1]

No	Cause	System Response
1	Student status is active	Show error message.

Relationships: N/A

Business Rules:

+ Only delete student has 'inactive' status;

+ Only delete when student has not been graded (no scores were recorded in the system).

2.2.5.13. <Super Admin, Class Admin>Add students in system(UC_78)

Use Case Diagram

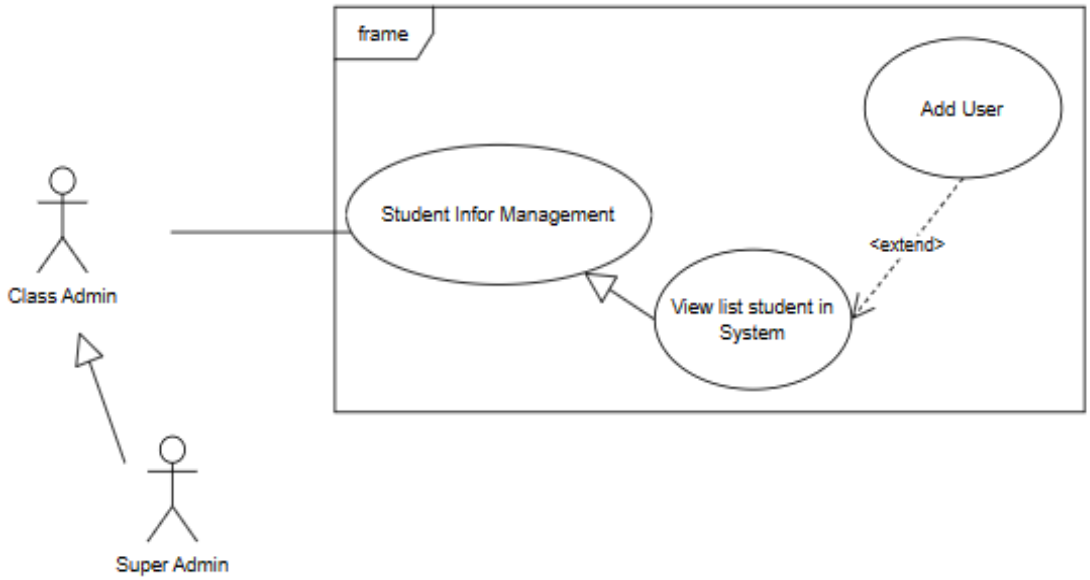


Figure 87 - <Super Admin, Class Admin>Add Student System

Use Case Specification

USE CASE – UC_78			
Use Case No.	78	**Use Case Version**	2.0
Use Case Name	Add Student In System		
Author	Nguyễn Mạnh Cường		
Date	20/3/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin.

Summary:

- Allow user to add the information of each student.

Goal:

- Add the new information of each student.

Triggers:

- User click on "Add student" button.

Preconditions:

- User at "Student List In A Class Screen"

Post Conditions:

- **Success:** User add student's information successfully.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Add new" button.	The system display add new student system page
2	User click on "Save" icon to add the student's information.	The system change the text to form input for admin to add information.

Alternative: N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1	"Name" is empty	Focus in "Name" field. Show error message
2	"Phone" is empty	Focus in "Phone" field. Show error message
3	"Email" is empty	Focus in "Email" field. Show error message

4	“RECer” is empty	Focus “RECer” field. Show error message

[Exceptions 2]

No	Cause	System Response
1	Email is not contain @.	Show error message.
2	The first number of phone number is different with 0.	Show error message.
3	Phone number length different with 10.	Show error mesage

Relationships: N/A

Business Rules:

- Email: Contain “@”.
- Phone number must start with 0.
- Phone number length must be equal 10

2.2.5.14. <Super Admin, Class Admin>Add students in class (UC_79)

Use Case Diagram

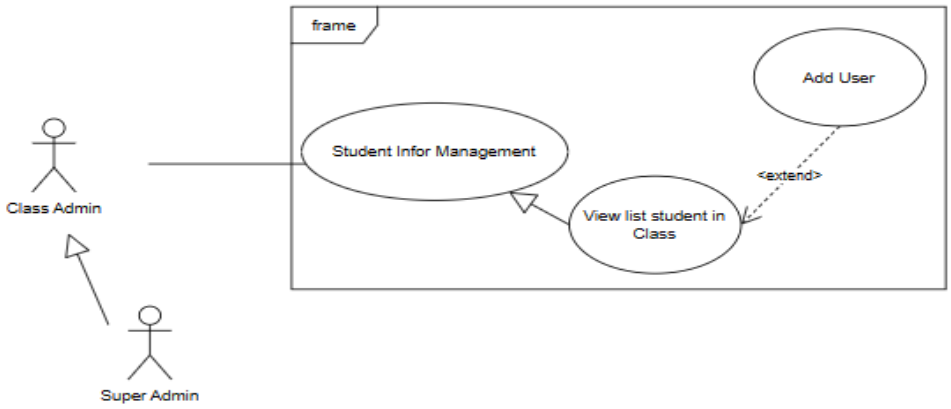


Figure 88 - <Super Admin, Class Admin >Add Student In Class

Use Case Specification

USE CASE – UC_79			
Use Case No.	79	**Use Case Version**	2.0
Use Case Name	Add Student In Class		
Author	Nguyễn Mạnh Cường		
Date	20/3/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin.

Summary:

- Allow user to add the information of each student.

Goal:

- Add the new information of each student.

Triggers:

- User click on "Add student" button.

Preconditions:

- User at "Student List In A Class Screen"

Post Conditions:

- **Success:** user add student's information successfully.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Add new" button.	The system display add new student in class page
2	User click on "Save" icon to add the student's information.	The system change the text to form input for admin to add information.

Alternative: N/A

Exceptions:
[Exception 1]

No	Cause	System Response
1	"Name" is empty	Focus in "Name" field. Show error message
2	"Phone" is empty	Focus in "Phone" field. Show error message
3	"Email" is empty	Focus in "Email" field. Show error message
4	"RECer" is empty	Focus "RECer" field. Show error message

[Exceptions 2]

No	Cause	System Response
1	Email is not contain @.	Show error message.
2	The first number of phone number is different with 0.	Show error message.
3	Phone number length different with 10.	Show error mesage

Relationships: N/A

Business Rules:

- Email: Contain "@".
- Phone number must start with 0.

- Phone number length must be equal 10

2.2.5.15. <Super Admin, Class Admin> Add Class Information (UC_80)

Use Case Diagram

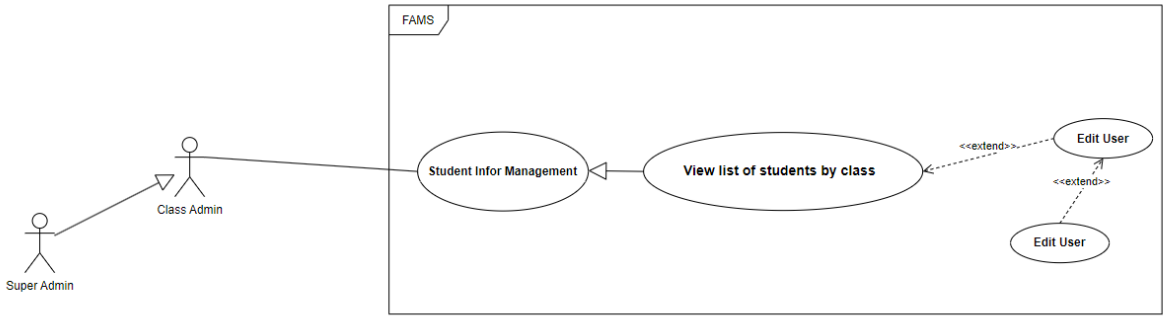


Figure 89 - <Super Admin, Class Admin> Add Class Information

Use Case Specification

USE CASE – UC_80			
Use Case No.	80	**Use Case Version**	2.0
Use Case Name	Add Class Information		
Author	Nguyễn Sơn Lam		
Date	10/04/2023	**Priority**	Normal

Actor:
- Super Admin, Class Admin

Summary:
- This use case allows actors add a student into a class.

Goal:
- Create new class information for student.

Triggers:
- Actors click on "Create" button.

Preconditions:
- Actors at "Edit Student"

Post Conditions:
- **Success:** Display success message and new class information will be display in edit student screen.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Guest click on ⊕ button.	The system request information in form: - Class: Dropdown.
2	Guest inputs information.	System validate the information and show succesfull message.

Alternative: N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1	"Class" is empty	Focus in "Class" field. Show error message

Relationships: N/A

Business Rules: N/A

2.2.6. Score Management Over View Use Case

Use Case Diagram

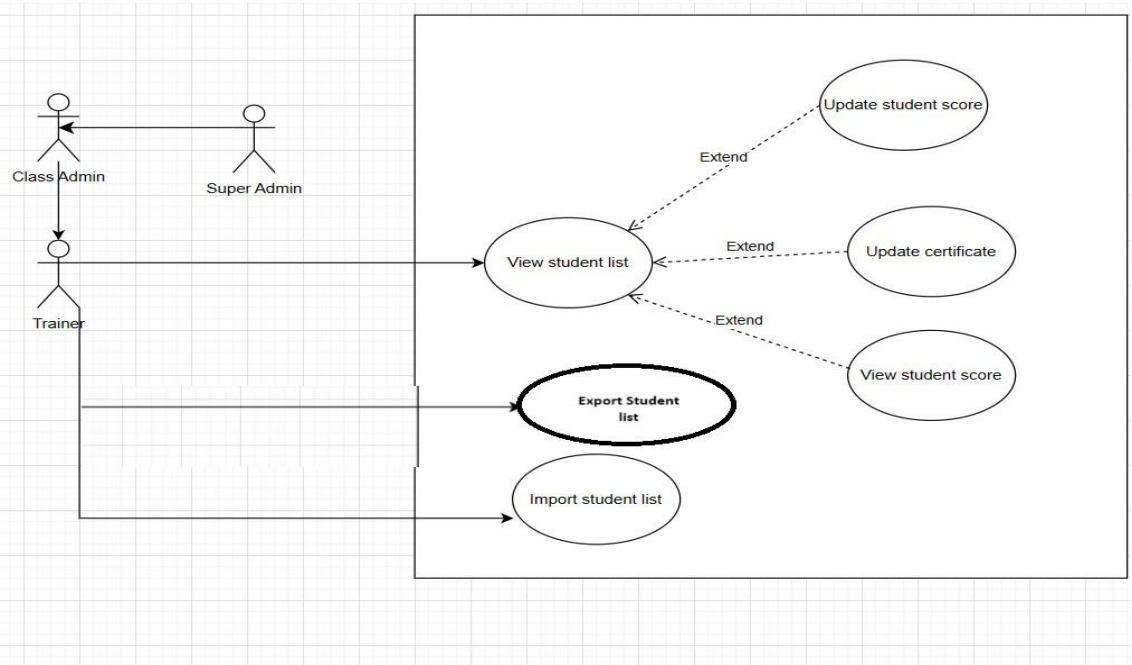


Figure 90 - Score Management Over View Use Case

2.2.6.1. <Super Admin, Class Admin, Trainer> Import list of trainee's score (UC_81)

Use Case Diagram

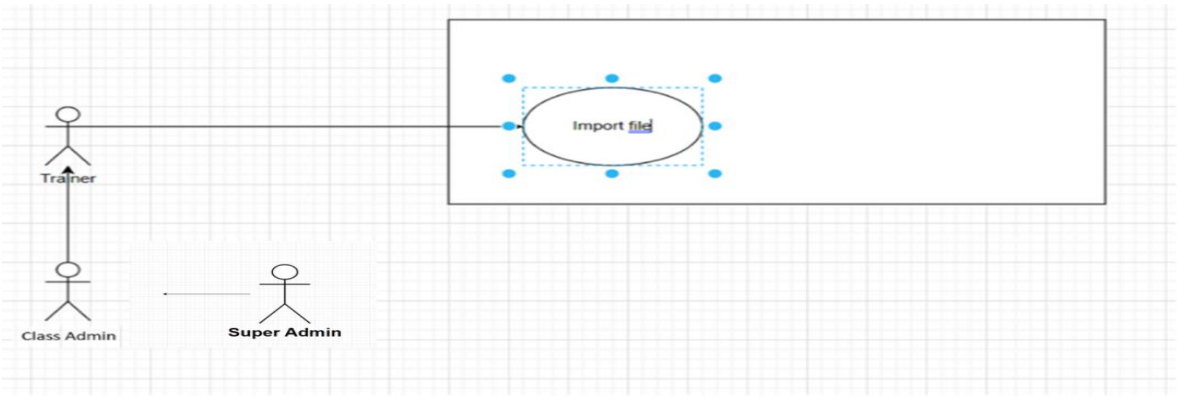


Figure 91 - <Super Admin, Class Admin, Trainer> Import list of trainee's score

Use Case Specification

USE CASE – US_81			
Use Case No.	81	**Use Case Version**	1.0
Use Case Name	Import list of trainee's score		
Author	Nguyễn Kỳ Anh Minh		
Date	13/3/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin, Trainer.

Summary:

- This use case allows trainer or class admin to import list of trainee's score.

Goal:

- Student score list will be imported into database.

Triggers:

- User click on "Import Score" button.

Preconditions:

- User at "Syllabus page".

Post Conditions:

- **Success:** An import successfully message will show up.
- **Fail:** Show import error message.

Main Success Scenario:

Step	Actor Action	System Response

1	Guest click on “Select” button.	The system request information in form: - File: Select file to import. - Import template: Dowload template.
2	Guest click on import.	- System validate the information and show succesfull message if not duplicate - If found duplicate will allow choice one from three option (allow, replace, skip) then click “Import” button again.

Alternative: N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1	“File” is empty	Focus in “File” field. Show error message

[Exceptions 2]

No	Cause	System Response
1	File is not Excel file	Show error message

Exceptions 3]

No	Cause	System Response
1	File is not match format	Show error message

Exceptions 4]

No	Cause	System Response
1	Field in file missing or null	Show error message

Exceptions 5]

No	Cause	System Response
1	Datatype in field in file is wrong	Show error message

Relationships: N/A

Business Rules:

- File: The import file must be excel file.

2.2.6.2. <Super Admin, Class Admin, Trainer> Export list of trainee's score (UC_82)

Use Case Diagram

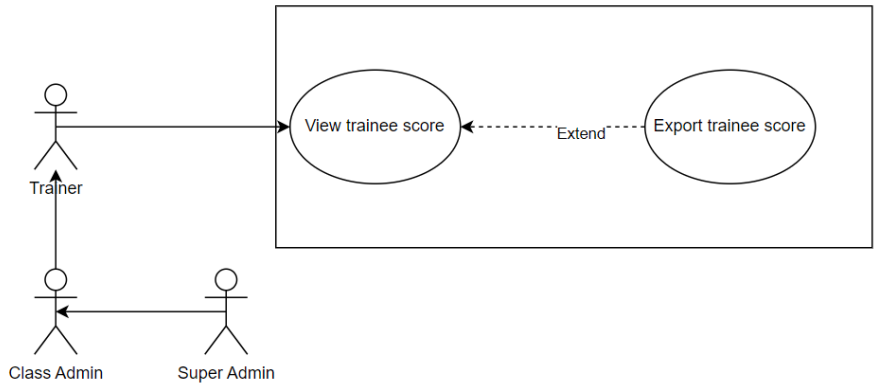


Figure 92 - <Super Admin, Class Admin, Trainer> Export list of trainee's score

Use Case Specification

USE CASE – US_82			
Use Case No.	82	**Use Case Version**	1.0
Use Case Name	Export list of trainee's score		
Author	Nguyễn Đức Anh		
Date	14/6/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin, Trainer.

Summary:

- This use case allows trainer or class admin to export list of trainee's score.

Goal:

- Student score list will be imported into database.

Triggers:

- User click on “export Score” button.

Preconditions:

- User at “Syllabus page”.

Post Conditions:

- **Success:** An export will be download.
- **Fail:** Show export error message.

Main Success Scenario:

Alternative: N/A

Exceptions:

Relationships: N/A

Business Rules: N/A

2.2.6.3. <Super Admin, Class Admin, Trainer> Update the score list for each student (UC_83)

Use Case Diagram

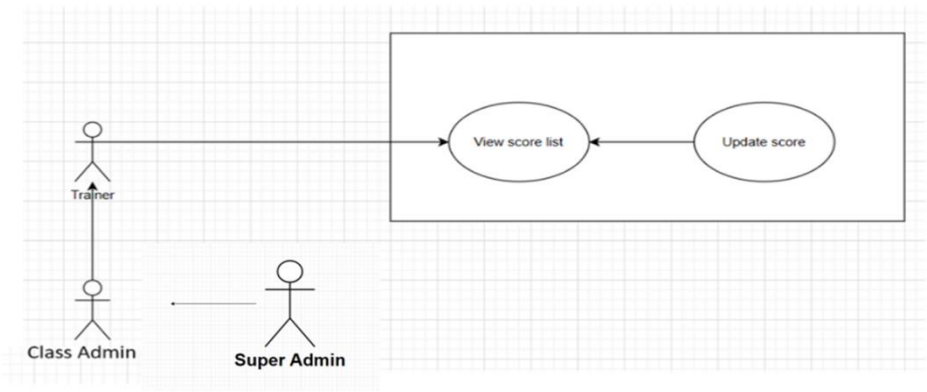


Figure 93 - <Super Admin, Class Admin, Trainer>Update the score list for each student.

Use Case Specification

USE CASE – US_83			
Use Case No.	83	**Use Case Version**	1.0
Use Case Name	Update the score list for each student.		

Author	Nguyễn Đức Anh		
Date	13/3/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin, Trainer.

Summary:

- This use case allows Admin to update the score list for each student.

Goal:

User can overview the updated score of each student.

Triggers:

- User click on score update score button.

Preconditions:

- User at "View trainee's score page".

Post Conditions:

- **Success:** A trainee's final studying result page will show up.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click "Update".	The scores have been successfully updated." will be showed
Step	Actor Action	System Response

Alternative: N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1	User input score is not between 1-100	Show invalid message

Relationships: N/A

Business Rules: N/A

2.2.6.4. <Super Admin, Class Admin, Trainer> View the trainee's score, ranking and final assessment of all modules per class (UC_84)

Use Case Diagram

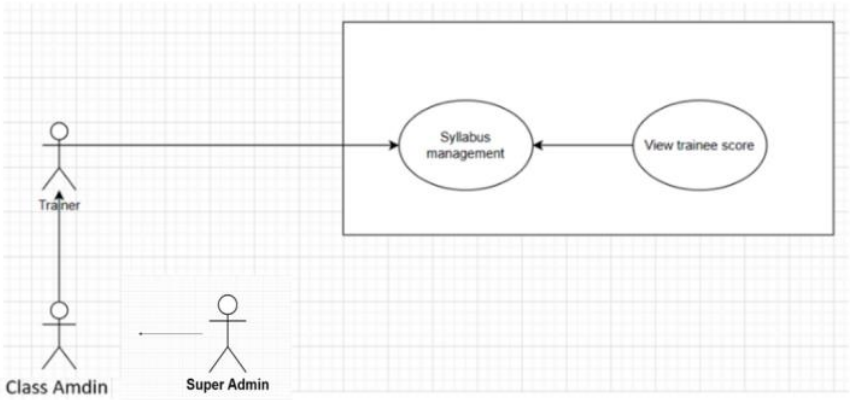


Figure 94 - <Super Admin, Class Admin, Trainer>View the trainee's score, ranking and final assessment of all modules per class

Use Case Specification

USE CASE – US_84			
Use Case No.	84	**Use Case Version**	1.0
Use Case Name	View the trainee's score.		
Author	Nguyễn Minh Quang		
Date	13/3/2024	**Priority**	Normal
Actor: - Super Admin, Class Admin, Trainer. **Summary:** - This use case allows Admin to view the trainee's score, ranking and final assessment of all modules per class. **Goal:** - Trainee's score, ranking and final assessment of all modules per class will be shown. **Triggers:** - User select the select a specific class in Syllabus Management then select Score			

Management tab.

Preconditions:

- User at "View trainee's score page".

Post Conditions:

- **Success:** A trainee's score page will show up.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click "Syllabus Management".	The system show Syllabus Management page.
2	Admin click "View Trainee Score" button.	The system show the relative class score table.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Admin can see all trainee score in a specific class.

2.2.6.5. <Super Admin, Class Admin, Trainer> View final studying result per trainee (UC_85)

Use Case Diagram

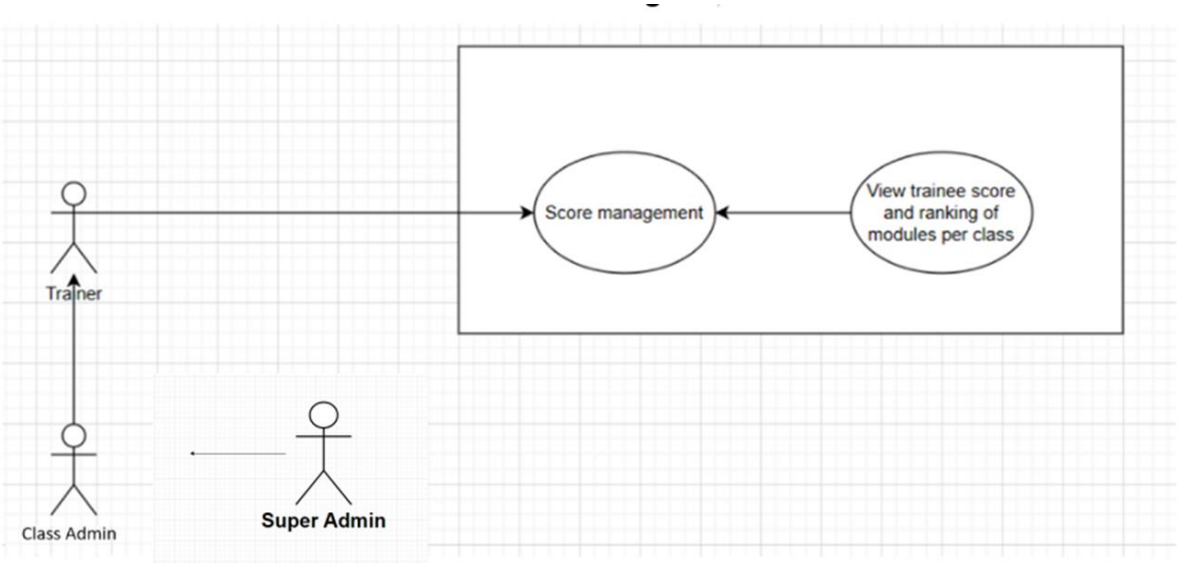


Figure 95 - <Super Admin, Class Admin, Trainer> View the studying result record per trainee.

Use Case Specification

USE CASE – US_85			
Use Case No.	85	**Use Case Version**	1.0
Use Case Name	View the studying result record per trainee.		
Author	Nguyễn Hiệp Phú		
Date	13/3/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin, Trainer.

Summary:

- This use case allows Admin to to view final studying result per trainee.

Goal:

User can follow-up the studying status of each trainee.

Triggers:

- User click on score editing button.

Preconditions:

- User at "View trainee's score page".

Post Conditions:

- **Success:** A trainee's final studying result page will show up.

- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User access View trainee's score page.	The system show information in table: The Score Management is displayed with information of scores and ranking of all students belongs to this class.
2	Guest click on edit score button.	System redirect user to score page.

Alternative: N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1		

Relationships: N/A

Business Rules: N/A

2.2.6.6. <Super Admin, Class Admin, Trainer> Update status of certificate award (UC_86)

Use Case Diagram

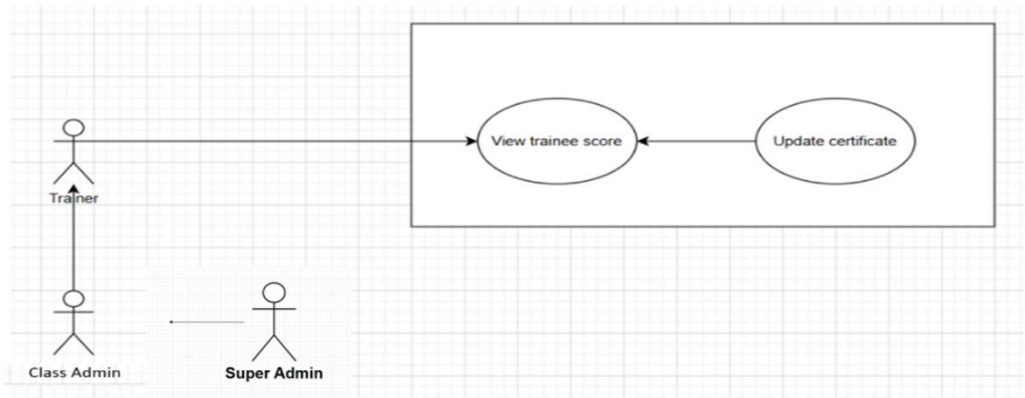


Figure 96 - <Super Admin, Class Admin, Trainer>Update status of certificate award

Use Case Specification

USE CASE – US_86			
Use Case No.	86	**Use Case Version**	1.0
Use Case Name	Update status of certificate award		
Author	Đinh Thế Vinh		
Date	13/3/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin, Trainer.

Summary:

- This use case allows Admin to update status of certificate award.

Goal:

Admin can know which students already receive the certificate and which not yet.

Triggers:

- User click on score update certificate button.

Preconditions:

- User at "View trainee's score page".

Post Conditions:

- **Success:** A update successfully message will show up.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User access View trainee's score page.	The system show information in table: The Score Management is displayed with information of scores and ranking of all students belongs to this class.
2	Guest click on update certificate button.	If all actions succeed, then a successful toast message will pop up, otherwise an error message will appear.

Alternative: N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1	User input score is an invalid number	Show error in toast message
2	User input invalid email	Show error in toast message
3	User input invalid phone number	Show error in toast message
4	User input an existing student ID	Show error in toast message

Relationships: N/A

Business Rules: None

2.2.7. Reservation Management Over View Use Case

Use Case Diagram

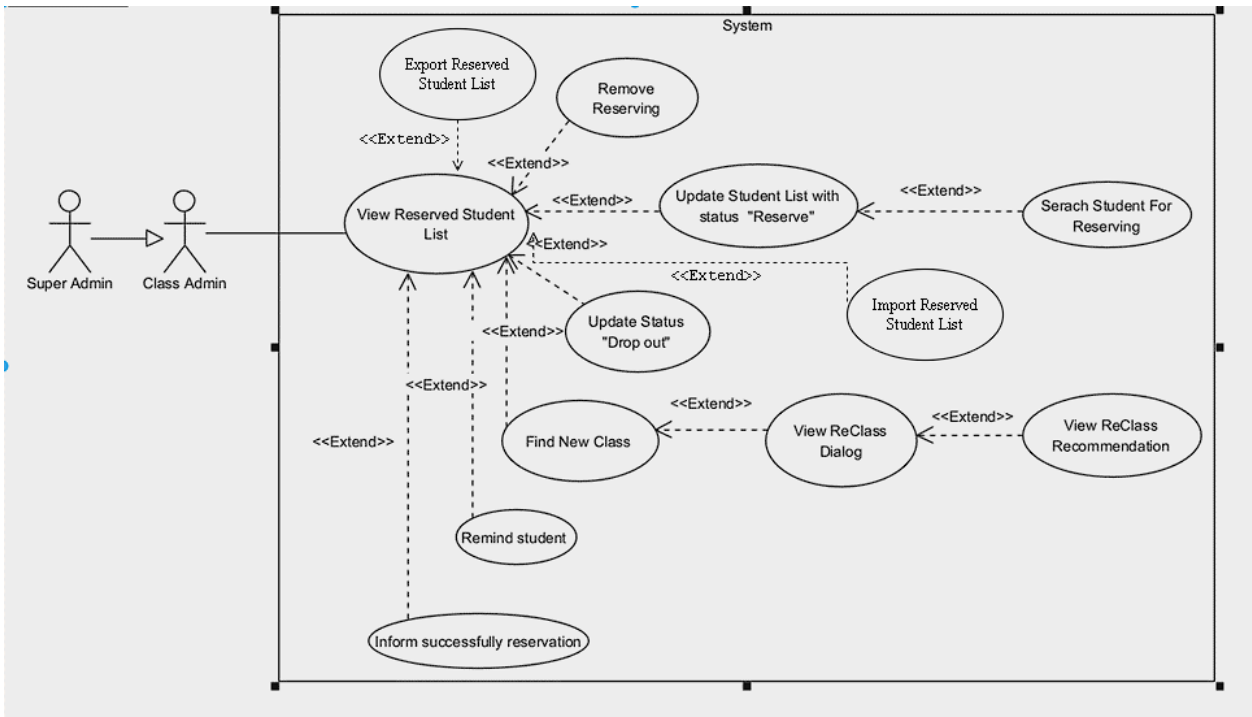


Figure 97 - Reservation Management Over View Use Case

2.2.7.1. <Super Admin, Class Admin> Update list of students with status "Reservation" (UC_87)

Use case Diagram

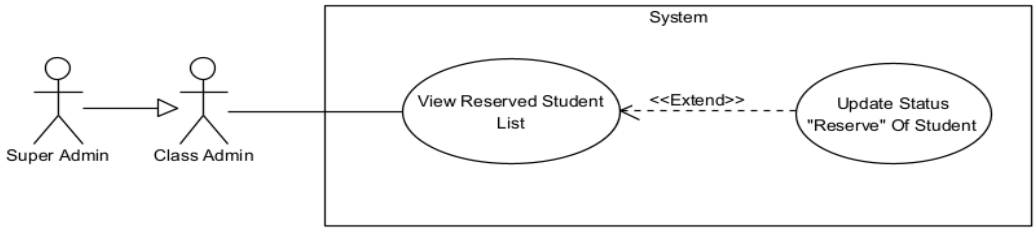


Figure 98 - <Super Admin, Class Admin> Update list of students with status "Reservation"

Use Case Specification

USE CASE – UC_87			
Use Case No.	87	**Use Case Version**	2.0
Use Case Name	Update list of students with status "Reservation"		
Author	Nguyễn Minh Phúc		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super admin, class admin

Summary:

- This use case allows the actor to update list of students with status "Reservation"

Goal:

- Actor will update list of students with status "Reservation"

Triggers:

- Actor clicks on the "Add reserving" button.

Preconditions:

- Actor is at "Reserved Student List Page".

Post Conditions:

- **Success:** System shows the successful message and adds new students into reserved students list.
- **Fail:** System shows the error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Actor clicks on the "Add reserving" button.	The system displays a dialogue "Add reserving" for the user to fill in the information: - Select student: input. - Class name: input. - Class code: input. - Current module: input. - Reserving reason: input. - Reserving period: date time. - Reserving conditions: checkbox. - Active reserving: toggle button. - Cancel: button. - Create: button.
2	Actor input Email or StudentId into Selected	- The system searches and displays all classes students are studying at

	student	Class name.
3	Actor chooses one class to reserve.	- The system will autofill Current module and class code.
4	Actor inputs reserving period, reason, checkbox condition then click on the "Create" button.	- The system creates new students and pop up successful messages.

Alternative:N/A

Exceptions:

No	Cause	System Response
1	The Email or StudentID does not exist.	System shows a "There are no students available" message.
2	Actor does not fill the text fields.	System shows errors under each of the text fields.

Relationships: View reserved student list

Business Rules:

- The Email or StudentID must exist.
- The End Date of Reserving Period must be larger than the Start Date.
- The student must be studying in at least one class.

2.2.7.2. <Super Admin, Class Admin> Terminate/ Remove status "Reservation" (UC_88)

Use case Diagram

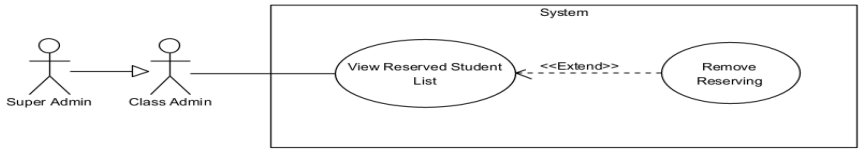


Figure 99 - <Super Admin, Class Admin> Terminate/ Remove status "Reservation"

Use Case Specification

USE CASE – UC_88			
Use Case No.	88	**Use Case Version**	2.0
Use Case Name	Terminate/Remove status "Reservation"		
Author	Nguyễn Minh Phúc		
Date	20/03/2024	**Priority**	Normal

Actor:

- Super admin, class admin

Summary:

- This use case allows the actor to remove the record of a student's reservation.

Goal:

- Actor will remove the record of a student's reservation.

Triggers:

- Actor clicks on the "Remove reserve" button.

Preconditions:

- Actor at "Reserved Student List Page"
- The current time has not exceeded the period of reserved class.
- The status of the student must be "Reserve".

Post Conditions:

- **Success:** System changes status of student from "Reserve" to "In class" and this student will be removed from Reserved Student List.
- **Fail:** Show "Something wrong" pop-up message.

Main Success Scenario:

Step	Actor Action	System Response
1	On the "Reservation Student List" page, the actor clicks on the "Pop-up menu" icon to display the Sub-action Board.	The Sub-action Board has four buttons: - Re Class. - Remind. - Drop out. - Remove reserve.

2	Actor clicks on the “Remove reserve” button in the Sub-action Board.	The system shows a confirmation dialog for removing the reservation with “Do you really want to remove this student from reserving class” notification and has two button which are “Cancel” and “Delete”.
3	Actor clicks on the “Delete” button.	The system changes the status of this student to “In class” and this student will be removed from the reserved list.

Alternative:

Step	Actor Action	System Response
1	On the "Reservation Student List" page, the actor clicks on the "Pop-up menu" icon to display the Sub-action Board.	The Sub-action Board has four buttons: - Re Class. - Remind. - Drop out. - Remove reserve.
2	Actor clicks on the “Remove reserve” button in the Sub-action Board.	The system shows a confirmation dialog for removing the reservation with “Do you really want to remove this student from reserving class” notification and has two buttons which are “Cancel” and “Delete”.
3	Actor clicks on “Cancel” button.	The system redirects to the "Reservation Student List" page.

Exceptions:

N/A

Relationships: View reserved student list

Business Rules:

- The current time has not exceeded the period of reserved class.
- Reserved class status is not “Completed”.

• The status of the student must be “Reserve”.

2.2.7.3. <Super Admin, Class Admin> Find new class for student after “Reservation” (UC_89)

Use case Diagram

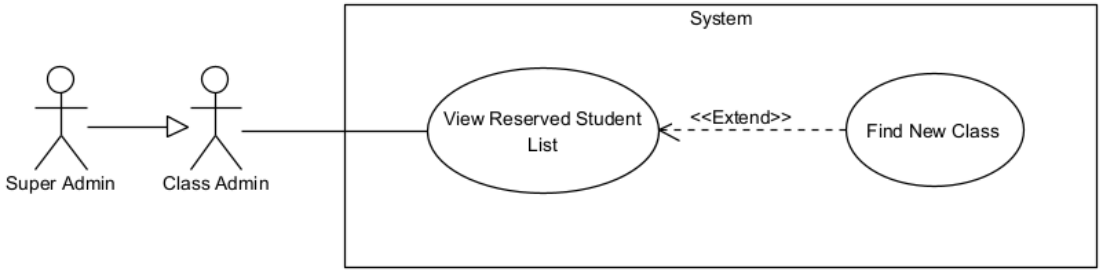


Figure 100 - <Super Admin, Class Admin> Find new class for student after “Reservation”

Use Case Specification

USE CASE – UC_89			
Use Case No.	89	**Use Case Version**	2.0
Use Case Name	Find new class for student after “Reservation”		
Author	Nguyễn Minh Phúc		
Date	20/03/2024	**Priority**	Normal
Actor: • Super admin, class admin **Summary:** • This use case allows actor to find an available new class for students so they can resume their studies after a break. **Goal:**			

- Actor will resume their studies after a break.

Triggers:

- Actor clicks on the “Re Class” button.

Preconditions:

- Actor at “Reserved Student List Page”
- The current time has exceeded the period of reserved class.
- The status of the student must be “Reserve”.

Post Conditions:

- **Success:** System backs to Re-Class Dialog and changes status to “In class”.
- **Fail:** Show “Something wrong” pop-up message.

Main Success Scenario:

Step	Actor Action	System Response
1	On the "Reservation Student List" page, the actor clicks on the "Pop-up menu" icon to display the Sub-action Board.	The Sub-action Board has four button: - Re Class. - Remind. - Drop out. - Remove reserve.
2	Actor clicks on the “Re Class” button in the Sub-action Board.	The system redirects to Re-Class dialog with information fields: - Class name: text. - Class code: text. - Class period: text. - Student score: text. - Current module: text. - Fee: text. - Quiz: table. - ASM: table. - Mock: table - Reserving information: text. - Reason: text - Conditions: textbox - Reclass possibility: text.

		- Recommended classes: cards.
3	Actor clicks on one of card of class in Recommended classes.	The system redirects to Re-Class recommendation with information fields: - Class information: text. - Class name: text. - Class code: text - Current modules: text. - Next modules: text. - Conditions: radio button. - Cancel: button. - Back to class: button.
4	Actor clicks on “Back to class” button.	The system changes the status of this student to “In class” and this student will not be removed from the reserved list.

Alternative:

Step	Actor Action	System Response
1	On the "Reservation Student List" page, the actor clicks on the "Pop-up menu" icon to display the Sub-action Board.	The Sub-action Board has four button: - Re Class. - Remind. - Drop out. - Remove reserve.
2	Actor clicks on the “Re Class” button in the Sub-action Board.	The system redirects to Re-Class dialog with information fields: - Class name: text. - Class code: text. - Class period: text. - Student score: text. - Current module: text. Fee: text. - Quiz: table.

		- ASM: table. - Mock: table - Reserving information: text. - Reason: text - Conditions: textbox - Reclass possibility: text. - Recommended classes: cards.
3	Actor clicks on one of the cards in Recommended classes.	The system redirects to Re-Class recommendation with information fields: - Class information: text. - Class name: text. - Class code: text - Current modules: text. - Next modules: text. - Conditions: radio button. - Cancel: button. - Back to class: button.
4	Actor clicks on the “Cancel” button.	The system redirects to the Re-Class dialog.

Exceptions:

N/A

Relationships: View reserved student list

Business Rules:

- The current time has exceeded the period of reserved class.
- Syllabus same as student's previous class before reserving
- Re-Class class status is not “Completed”.
- Reserved class status is “Completed”.
- The status of the student must be “Reserve”.

2.2.7.4. <Super Admin, Class Admin>Update status "Drop out" for student (UC_90)

Use case Diagram

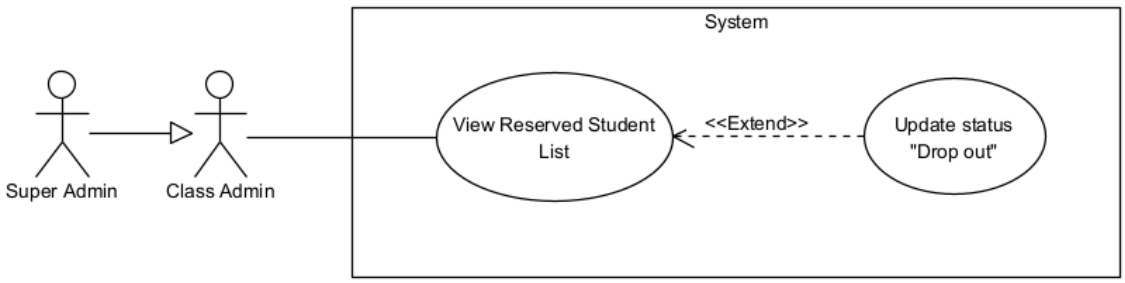


Figure 101 - <Super Admin, Class Admin> Update status "Drop out" for student

Use Case Specification

USE CASE – UC_90			
Use Case No.	90	**Use Case Version**	2.0
Use Case Name	Update status "Drop out".		
Author	Nguyễn Minh Phúc		
Date	20/03/2024	**Priority**	Normal
Actor: ● Super admin, class admin **Summary:** ● This use case allows the actor to update the status "Drop class" for the Attending 's reservation. **Goal:** ● Actor will change the status "Drop out" for the attending reservation who confirmed not to return to the class. **Triggers:**			

- Actor clicks on the “Drop out” button.

Preconditions:

- Actor at “Reserved Student List Page”
- The status of the student must be “Reserve”.

Post Conditions:

- **Success:** System changes status of student from “Reserve” to “Drop out” and this student will be removed from Reserved Student List.
- **Fail:** Show “Something wrong” pop-up message.

Main Success Scenario:

Step	Actor Action	System Response
1	On the "Reservation Student List" page, the actor clicks on the "Pop-up menu" icon to display the Sub-action Board.	The Sub-action Board has four buttons: - Re Class. - Remind. - Drop out. - Remove reserve.
2	Actor clicks on the “Drop out” button in the Sub-action Board.	The system shows a confirmation dialog for removing the reservation with “Do you really want to drop out this student from reserving class” notification and has two buttons which are “No” and “Yes”.
3	Actor clicks on the “Yes” button.	The system changes the status of this student to “Drop out” and this student will be removed from the reserved list.

Alternative:

Step	Actor Action	System Response
1	On the "Reservation Student List" page, the actor clicks on the "Pop-up menu" icon to display the Sub-action Board.	The Sub-action Board has four buttons: - Re Class. - Remind. - Drop out. - Remove reserve.
2	Actor clicks on the “Drop	The system shows a confirmation dialog

	out" button in the Sub-action Board.	for removing the reservation with "Do you really want to drop out this student from reserving class" notification and has two buttons which are "No" and "Yes".
3	Actor clicks on the "No" button.	The system redirects to the "Reservation Student List" page.

Exceptions: N/A

Relationships: View reserved student list

Business Rules:

- The status of the student must be "Reserve".

2.2.7.5. <Super Admin, Class Admin>View list of students with status "Reservation" (UC_91)

Use case Diagram

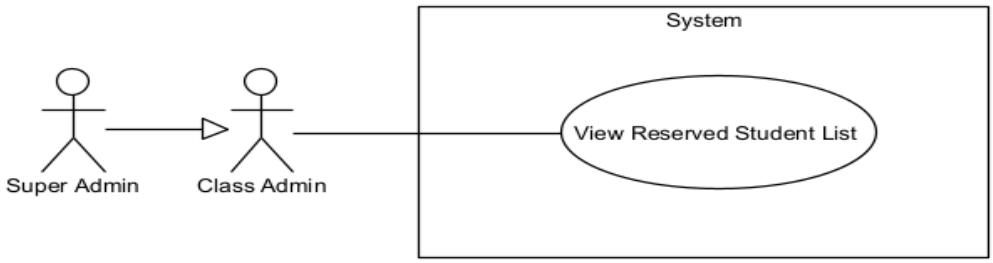


Figure 102 - <Super Admin, Class Admin> View list of student with status "Reservation"

Use Case Specification

USE CASE – UC_91			
Use Case No.	91	**Use Case Version**	2.0
Use Case Name	View list reservation		
Author	Nguyễn Sơn Lam		
Date	20/3/2023	**Priority**	Normal

Actor:

- Super Admin, Class Admin.

Summary:

- Allow actor to view list student with status "Reservation".

Goal:

- Display student having "Reservation" status.

Triggers:

- Admin click on "Reserve list" tab in sidebar.

Preconditions:

- Admin have already logged in.

Post Conditions:

- **Success:** Admin view reservation list successfully.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Actor click on "Reserve list" tab.	The system display reservation list.

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.7.6. <Super Admin, Class Admin>Export the information of each student from reserve list in system (UC_92)

Use Case Diagram

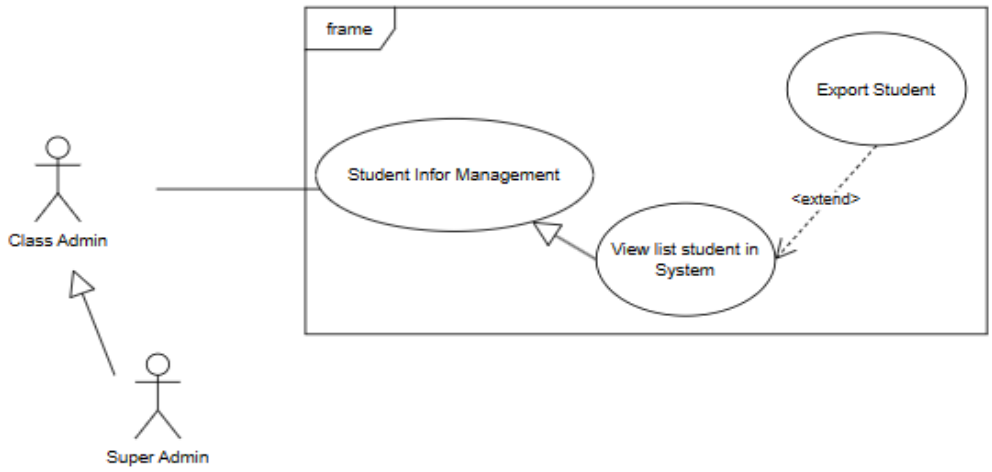


Figure 103 - <Class Admin, Super Admin>Export Data Excel In System

Use Case Specification

USE CASE – UC_92			
Use Case No.	92	**Use Case Version**	2.0
Use Case Name	Export Student List By System		
Author	Nguyễn Đức Linh		
Date	20/3/2024	**Priority**	Normal
Actor: ● Super Admin, Class Admin. **Summary:** ● Allow user to export the information of each student. **Goal:** ● Export exact information of each student show on reserve list Screen. **Triggers:** ● User click on “Export” button. **Preconditions:** ● User at “A Reserve List Screen” **Main Success Scenario:**			

Step	Actor Action	System Response
1	User click on " Export " button.	File ReservatedList.xlxs will be dowload automatically

Alternative: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

2.2.7.7. <Super Admin, Class Admin>Import reserve list (UC_93)

Use Case Diagram

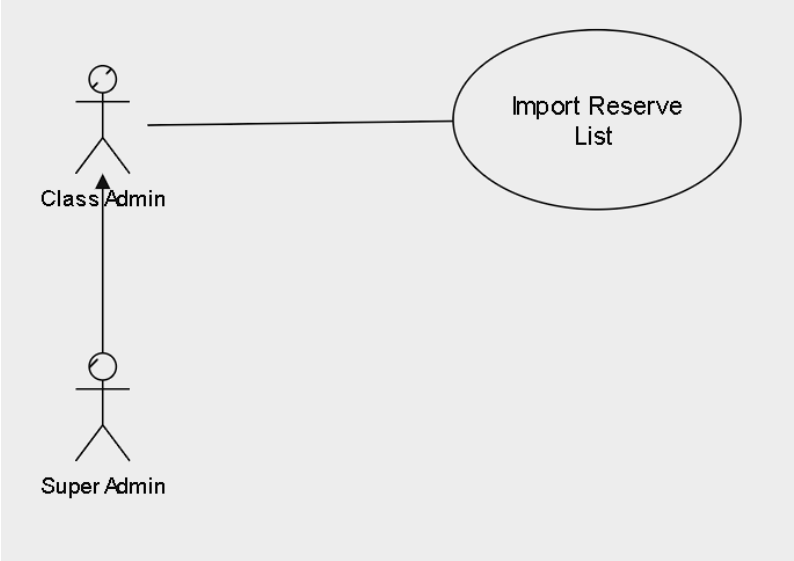


Figure 104 - <Class Admin, Super Admin> Import Data Excel In System

Use Case Specification

USE CASE – UC_93			
Use Case No.	**93**	**Use Case Version**	1.0
Use Case Name	Import Reserve List		
Author	Nguyễn Đức Linh		
Date	19/03/2024	**Priority**	Normal

Actor:

- Super Admin, Class Admin

Summary:

- This use case allows user to import reserve list from excel file.

Goal:

- Add new Reserve List

Triggers:

- User click on "Import" button.

Preconditions:

- User must login with roles "Super Admin" or "Class Admin"

Post Conditions:

- **Success:** User can add new reserve list.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User click on "Reserve List" in the menu	Show a dropdown list
2	User click on "Import"	Show a form for user to import a file
4	User select choices of Reserve List	
5	Click "Import" button	Show success message. [Exception 1]

Alternative: N/A

Exceptions: N/A

No	Cause	System Response
1	**Status is invalid**	**Focus on "Status" field. Show error message**

Relationships: N/A

Business Rules: N/A

2.2.8. Email Inform/Remind Over View Use Case

Use Case Diagram

Figure 105 - Email Inform/Remind Over View Use Case

2.2.8.1. <Super Admin, Class Admin> Create email template list (UC_94)

Use Case Diagram

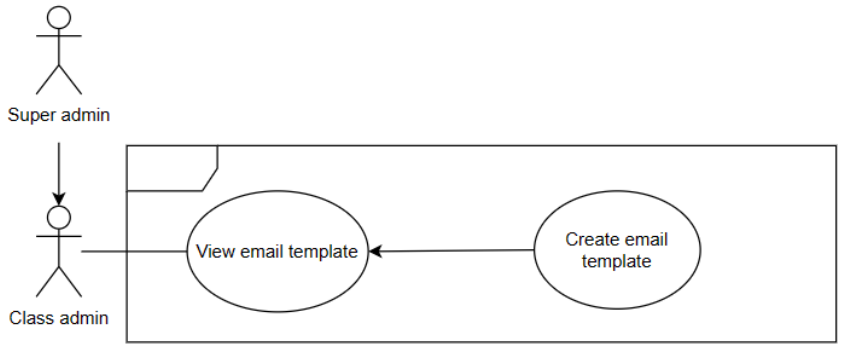


Figure 106 - <Super Admin, Class Admin> Create new email template

Use Case Specification

USE CASE – UC_94			
Use Case No.	94	**Use Case Version**	1.0
Use Case Name	Create email template		
Author	Nguyễn Hoàng Thái		
Date	13/03/2024	**Priority**	Normal

Actor:

- Super Admin and Class Admin

Summary:

- This use case allows admin to add new email template.

Goal:

- Add new email template.

Triggers:

- Admin click on “Add new” button.

Preconditions:

- User must login with role Super Admin or Class Admin.
- Admin at Email template list page.

Post Conditions:

- **Success:** Admin added new email template.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click on "Add new" button.	The system request information in form: - Name Template: Free text input. - Categories: Select option. - Apply to: Select option. - Description: Free text input.
2	Admin input information of email template.	System check validation [Exception 1]
3	Admin click "Create" button.	System redirect admin to email template list page

Alternative: N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1	Admin doesn't input information	Focus in all field. Show error message with reasons
2	"Name Template" is empty	Focus in "Name Template" field. Show error message
3	"Categories" is empty	Focus in "Categories" field. Show error message
4	"Apply to" is empty	Focus in "Apply to" field. Show error message
5	"Description" is empty	Focus in "Description" field. Show error message

Relationships: N/A

Business Rules: N/A

2.2.8.2. <Super Admin, Class Admin> Update email template (UC_95)

Use Case Diagram

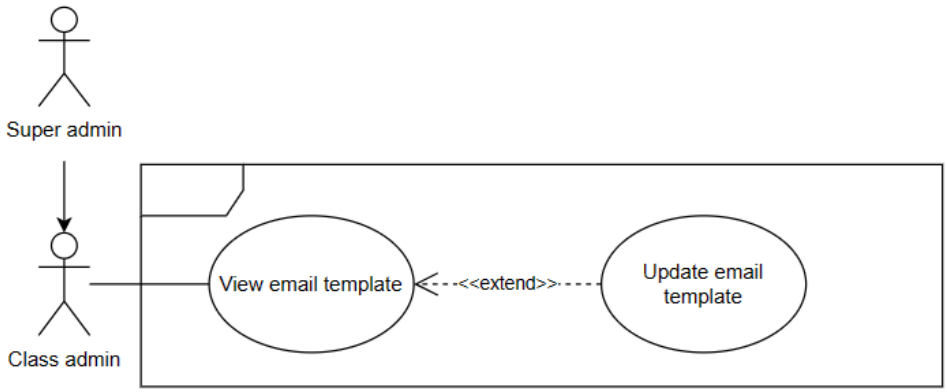


Figure 107 - <Super Admin, Class Admin> Update email template

Use Case Specification

USE CASE – UC_95			
Use Case No.	95	**Use Case Version**	1.0
Use Case Name	Update email template		
Author	Nguyễn Hoàng Thái		
Date	15/03/2024	**Priority**	Normal
Actor: - Super Admin and Class Admin **Summary:** - This use case allows admin to update email template. **Goal:** - Update email template. **Triggers:** - Admin click on button. **Preconditions:** - User must login with role Super Admin or Class Admin. - Admin at Email template list page. **Post Conditions:**			

- **Success:** Email template updated.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click on “From” field	The system show a list of sender for admin to choose.
2	Admin input new information	System check validation
3	Admin click on “Save” button	Show pop-up notification

Alternative: N/A

Exceptions:

No	Cause	System Response
1	“From” is empty	Focus in “From” field. Show error message
2	“Subject” is empty	Focus in “Subject” field. Show error message
3	“Body” is empty	Focus in “Body” field. Show error message

Relationships: N/A

Business Rules: N/A

2.2.8.3. <Super Admin, Class Admin> Search email template (UC_96)

Use Case Diagram

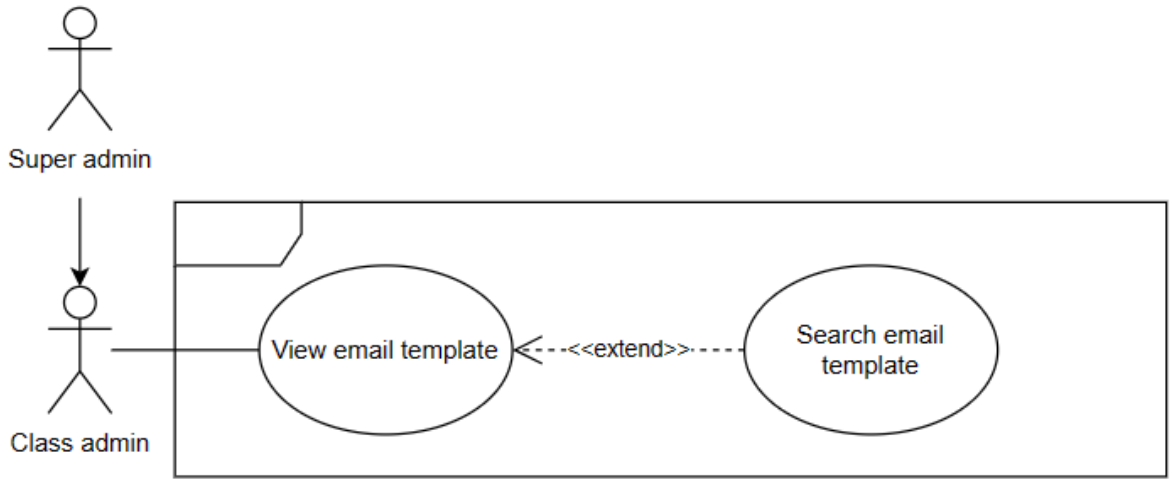


Figure 108 - <Super Admin, Class Admin> Search email template

Use Case Specification

USE CASE – UC_96			
Use Case No.	96	**Use Case Version**	1.0
Use Case Name	Search email template		
Author	Nguyễn Hoàng Thái		
Date	16/03/2024	**Priority**	Normal
Actor: - Super Admin and Class Admin **Summary:** - This use case allows admin to find email template. - Email template can be sorted. **Goal:** - Admin can find email template. **Triggers:** - Admin click on “Search by” bar. **Preconditions:** - User must login with role Super Admin or Class Admin.			

- Admin at Email template list page.

Post Conditions:

- **Success:** Email template with admin's condition will appear.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin input new information	System check validation [Exception 1]
2	Admin press Enter	Email template that requires searched information will appear [Exception 2]
3	Admin click on "Filter" button	The system request information in form: - Status: Select option. - Categories: Select option. - Apply to: Select option.

Alternative: N/A

Exceptions:

[Exceptions 1]

No	Cause	System Response
1	"Search by" bar is empty	Focus in "Template name" field. Show error message

[Exceptions 2]

No	Cause	System Response
1	No data with admin's condition	Show error message.

Relationships: N/A

Business Rules:

- Admin can quickly sort email template list by using tabs above search bar.

2.2.8.4. <Super Admin, Class Admin> Send email to trainer (UC_97)

Use Case Diagram

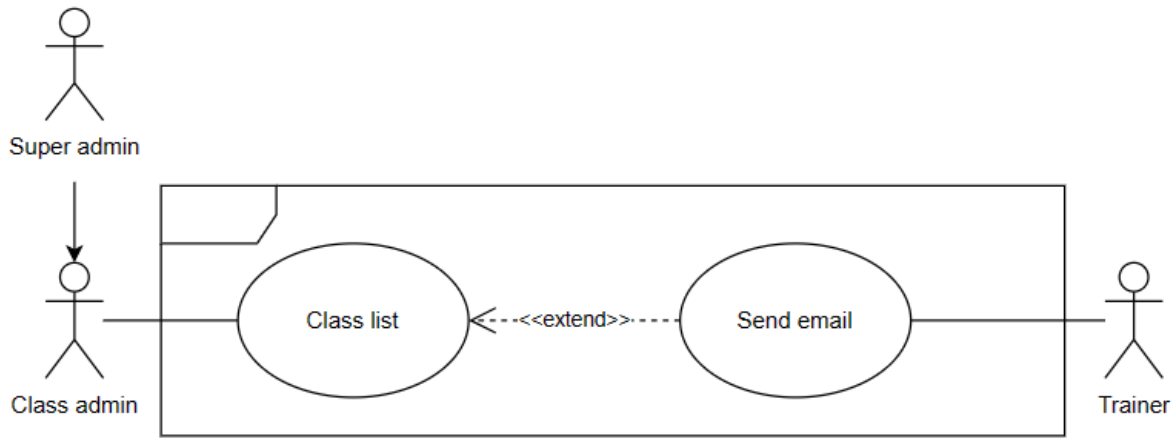


Figure 109 - <Super Admin, Class Admin> Update email template

Use Case Specification

USE CASE – UC_97			
Use Case No.	97	**Use Case Version**	1.0
Use Case Name	Send email		
Author	Nguyễn Hoàng Thái		
Date	13/03/2024	**Priority**	Normal
Actor: - Super Admin and Class Admin **Summary:** - This use case allows admin to send email to trainer. **Goal:** - Send email to trainer. **Triggers:** - Admin click on a trainer's name. **Preconditions:** - User must login with role Super Admin or Class Admin. - Admin at "Class list" page.			

Post Conditions:

- **Success:** Admin send an email to trainer.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click on trainer's name.	The system show a form with trainer's info and "Send Email" button.
2	Admin click on "Send Email" button on trainer info pop up.	The system show email template form with email detail and request information: - Template name: Select option
3	Admin click on "Preview" button on select template pop up.	The system show email preview form with email detail.
4	Admin click on "Send" button on preview pop up.	System check validation [Exception 2]

Alternative: N/A

Exceptions:

[Exceptions 1]

No	Cause	System Response
1	"Template name" is empty	Focus in "Template name" field. Show error message

[Exceptions 2]

No	Cause	System Response
1	Invalid email	Show error message.

Relationships: N/A

Business Rules: N/A

2.2.8.5. <Super Admin, Class Admin> Send email to reserved student (UC_98)

Use Case Diagram

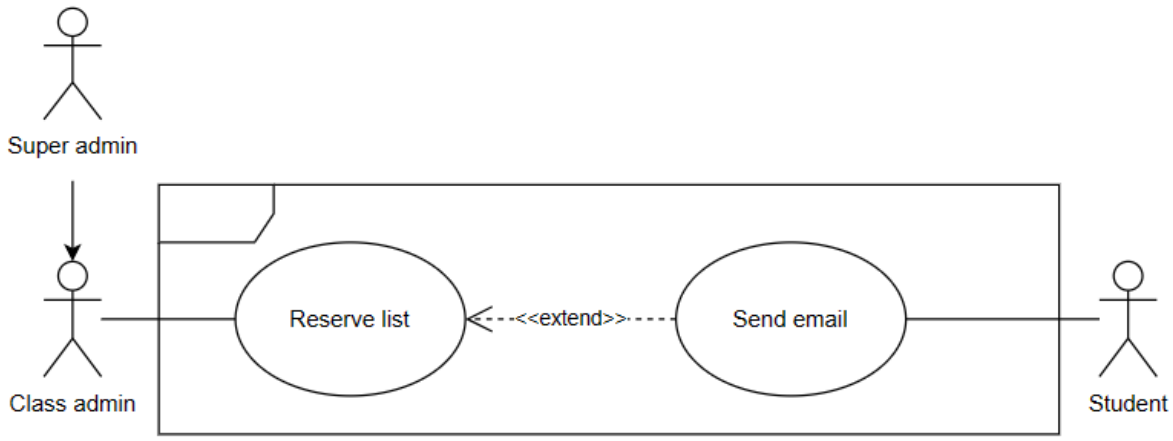


Figure 110 - <Super Admin, Class Admin> Send email to reserved student

Use Case Specification

USE CASE – UC_98			
Use Case No.	98	**Use Case Version**	1.0
Use Case Name	Send email		
Author	Nguyễn Hoàng Thái		
Date	13/03/2024	**Priority**	Normal
Actor: - Super Admin and Class Admin **Summary:** - This use case allows admin to send email using an email template to reserved student. **Goal:** - Send email to reserved student. **Triggers:** - Admin click on ••• button. **Preconditions:**			

- User must login with role Super Admin or Class Admin.
- Admin at "Reserve list" page.

Post Conditions:

- **Success:** Admin send an email to reserved student.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click ••• button.	The system show a pop up menu.
2	Admin click on "Remind" button on menu pop up.	The system show email template form with email detail and request information: - Template name: Select option
3	Admin click on "Preview" button on select template pop up.	The system show email preview form with email detail.
4	Admin click on "Send" button on preview pop up.	System check validation [Exception 2]

Alternative: N/A

Exceptions:

[Exceptions 1]

No	Cause	System Response
1	"Template name" is empty	Focus in "Template name" field. Show error message

[Exceptions 2]

No	Cause	System Response
1	Invalid email	Show error message.

Relationships: N/A

Business Rules: N/A

2.2.8.6. <Super Admin, Class Admin> Send email to student from class list (UC_99)

Use Case Diagram

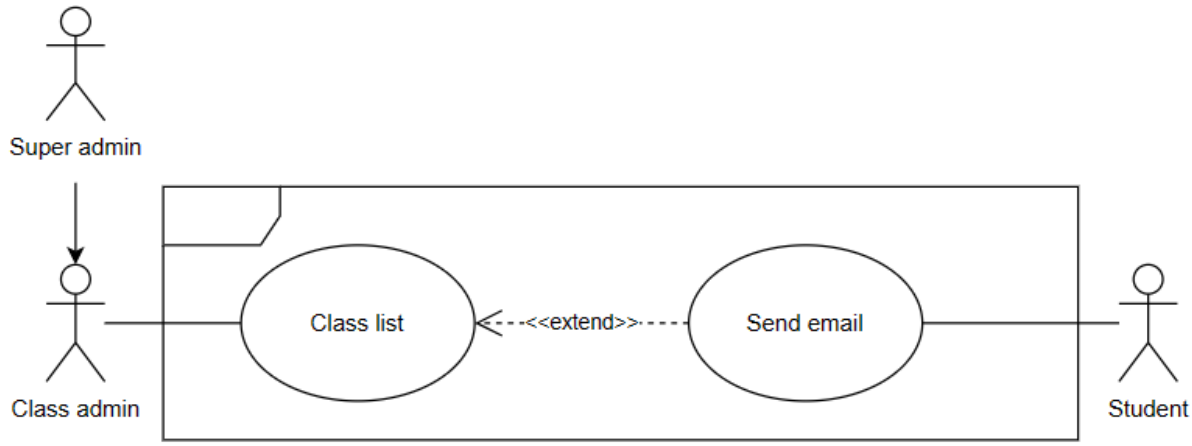


Figure 111 - <Super Admin, Class Admin> Send email to student

Use Case Specification

USE CASE – UC_99			
Use Case No.	99	**Use Case Version**	1.0
Use Case Name	Send email		
Author	Nguyễn Hoàng Thái		
Date	13/03/2024	**Priority**	Normal
Actor: - Super Admin and Class Admin **Summary:** - This use case allows admin to send email using an email template to student. **Goal:** - Send email to student. **Triggers:** - Admin click on ••• button in student list menu. **Preconditions:**			

- User must login with role Super Admin or Class Admin.
- Admin at "Class list" page.

Post Conditions:

- **Success:** Admin send an email to student.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click ••• button on student list.	The system show a pop up menu.
2	Admin click on "Send email" button on menu pop up.	The system show email template form with email detail and request information: - Template name: Select option
3	Admin click on "Preview" button on select template pop up.	The system show email preview form with email detail.
4	Admin click on "Send" button on preview pop up.	System check validation [Exception 2]

Alternative: N/A

Exceptions:

[Exceptions 1]

No	Cause	System Response
1	"Template name" is empty	Focus in "Template name" field. Show error message.

[Exceptions 2]

No	Cause	System Response
1	Invalid email	Show error message.

Relationships: N/A

Business Rules: N/A

2.2.8.7. <Super Admin, Class Admin> Send email to student from student list (UC_100)

Use Case Diagram

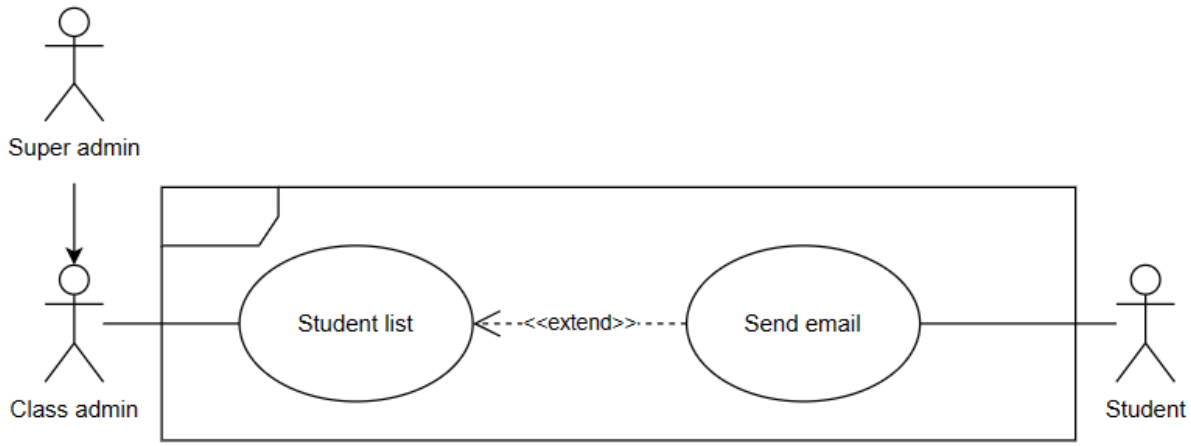


Figure 112 - <Super Admin, Class Admin> Send email to student

Use Case Specification

USE CASE – UC_100			
Use Case No.	100	**Use Case Version**	1.0
Use Case Name	Send email		
Author	Nguyễn Hoàng Thái		
Date	13/03/2024	**Priority**	Normal
Actor: - Super Admin and Class Admin **Summary:** - This use case allows admin to send email using an email template to student. **Goal:** - Send email to student. **Triggers:** - Admin click on ••• button in student list menu. **Preconditions:**			

- User must login with role Super Admin or Class Admin.
- Admin at "Student list" page.

Post Conditions:

- **Success:** Admin send an email to student.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click ••• button on student list.	The system show a pop up menu.
2	Admin click on "Send email" button on menu pop up.	The system show email template form with email detail and request information: - Template name: Select option
3	Admin click on "Preview" button on select template pop up.	The system show email preview form with email detail.
4	Admin click on "Send" button on preview pop up.	System check validation [Exception 2]

Alternative: N/A

Exceptions:

[Exceptions 1]

No	Cause	System Response
1	"Template name" is empty	Focus in "Template name" field. Show error message

[Exceptions 2]

No	Cause	System Response
1	Invalid email	Show error message

Relationships: N/A

Business Rules: N/A

2.2.8.8. <Super Admin, Class Admin> Export email template (UC_101)

Use Case Diagram

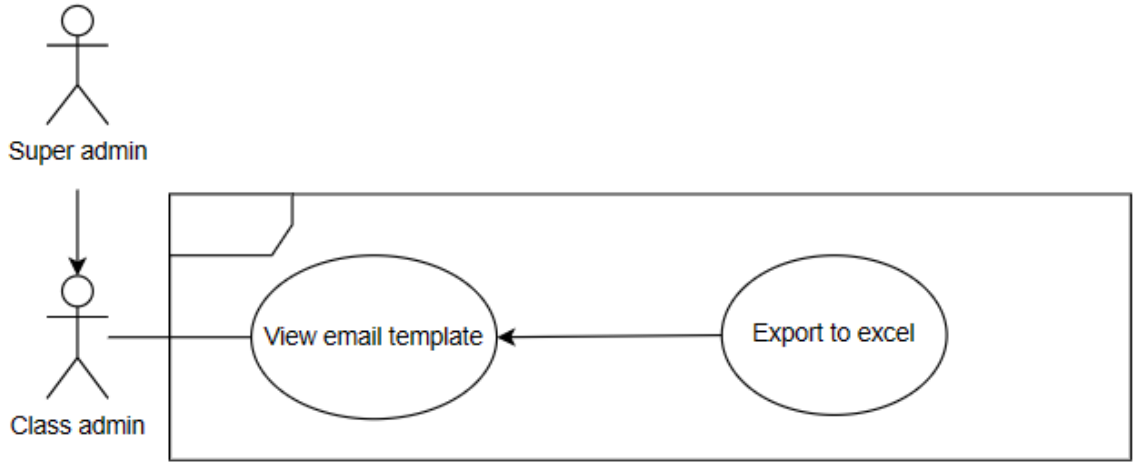


Figure 113 - <Super Admin, Class Admin> Search email template

Use Case Specification

USE CASE – UC_101			
Use Case No.	101	**Use Case Version**	1.0
Use Case Name	Search email template		
Author	Nguyễn Hoàng Thái		
Date	15/04/2024	**Priority**	Normal

Actor:
- Super Admin and Class Admin

Summary:
- This use case allows admin to export email template list to excel file and download automatically.

Goal:
- Admin can download excel file containing all email template.

Triggers:
- Admin click on "Export" button.

Preconditions:
- User must login with role Super Admin or Class Admin.
- Admin at Email template list page.

Post Conditions:
- **Success:** Excel file downloaded to Admin's pc.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin press Export button	System check validation [Exception 1]

Alternative: N/A

Exceptions:

[Exceptions 1]

No	Cause	System Response
1	Firewall block download	Show error message

Relationships: N/A

Business Rules: N/A

3. Software System Attribute

3.1. Usability

- User interface:
  - Provide a convenient way to interact with system.
  - Screen layout and arrangement are clear and easy to use.
  - Fields are grouping to different groups to help user observes easily.
- Usability
  - With simple and clear UI, user can learn how to use system quickly without any problems.

3.2. Reliability

- The data of project take from official sources
- Password is encoded by BCrypt

3.3. Availability

- Website must be available 24/7

3.4. Security

- Privacy: Each role of user has a specific permission to interact with system.
- System always checks authorization and authenticated before doing function of admin and user
- Only admin can grant permission to manage or change data of website.
- The password of user have been encrypted in database.

3.5. Maintainability

- The system is divided into separated modules.
- The code is easy to maintain and upgrade.

3.6. Portability

- The website is running on Windows XP or above.
- Providing easy installation.

3.7. Performance

- All request is handle in maximum time of 5 seconds.

4. Conceptual Diagram

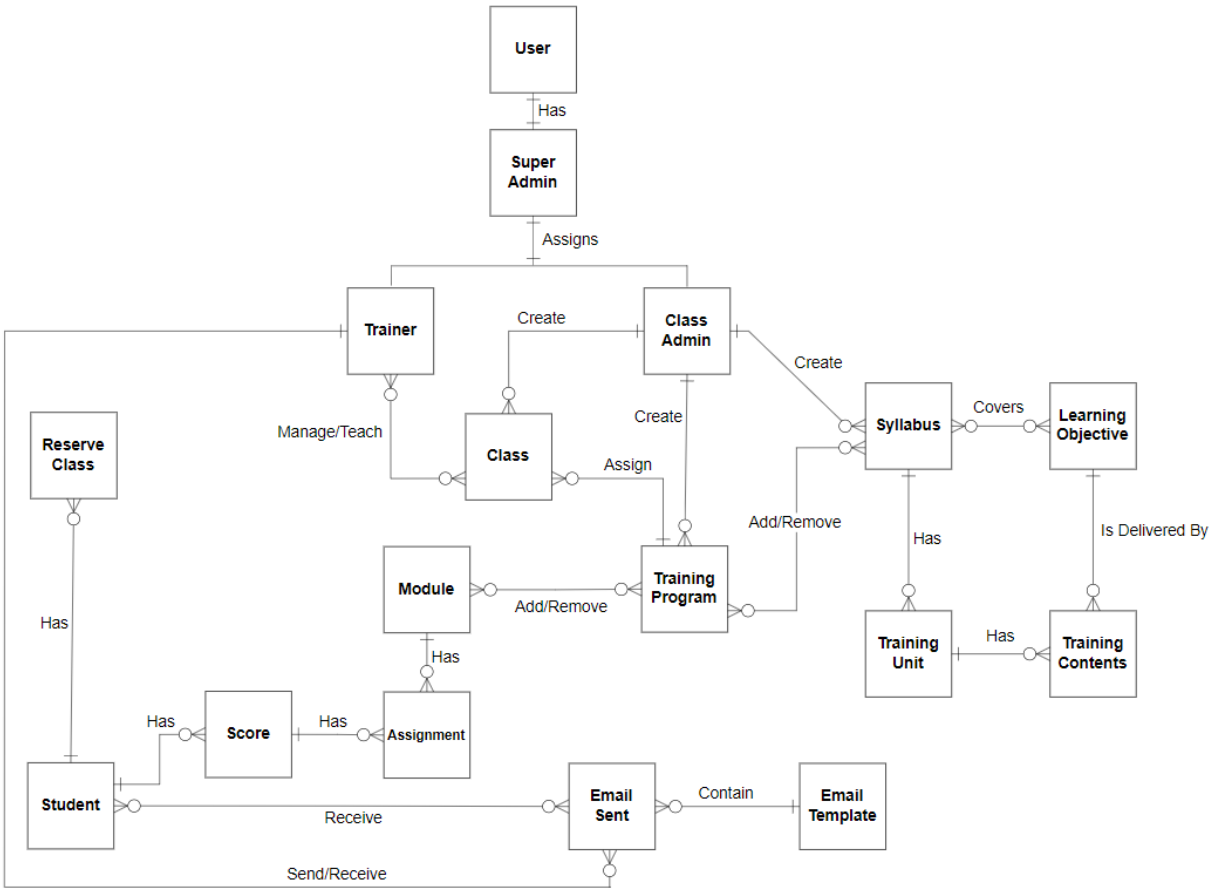


Figure 114 - Conceptual Diagram

Data Dictionary

Conceptual Data Dictionary: describe contents of all entities	
Conceptual Name	**Description**
User	Contain all information of user

Super Admin	Contain all information of super admin
Class Admin	Contain all information of class admin
Trainer	Contain all information of trainer
Class	Contain all information of class
Training Program	Contain all information of training program
Syllabus	Contain all information of syllabus
Learning Objective	Contain all information of learning objective
Training Contents	Contain all information of training contents
Training Unit	Contain all information of training unit
Reserve Class	Contain all information of reserve class
Student	Contain all information of student
Score	Contain all information of score
Assignment	Contain all information of assignment
Module	Contain all information of module
Email Sent	Contain all information of email sent
Email Template	Contain all information of email template

Table 9 - Conceptual Diagram Data Dictionary

D. Software Design Description

1. System Architectural Design

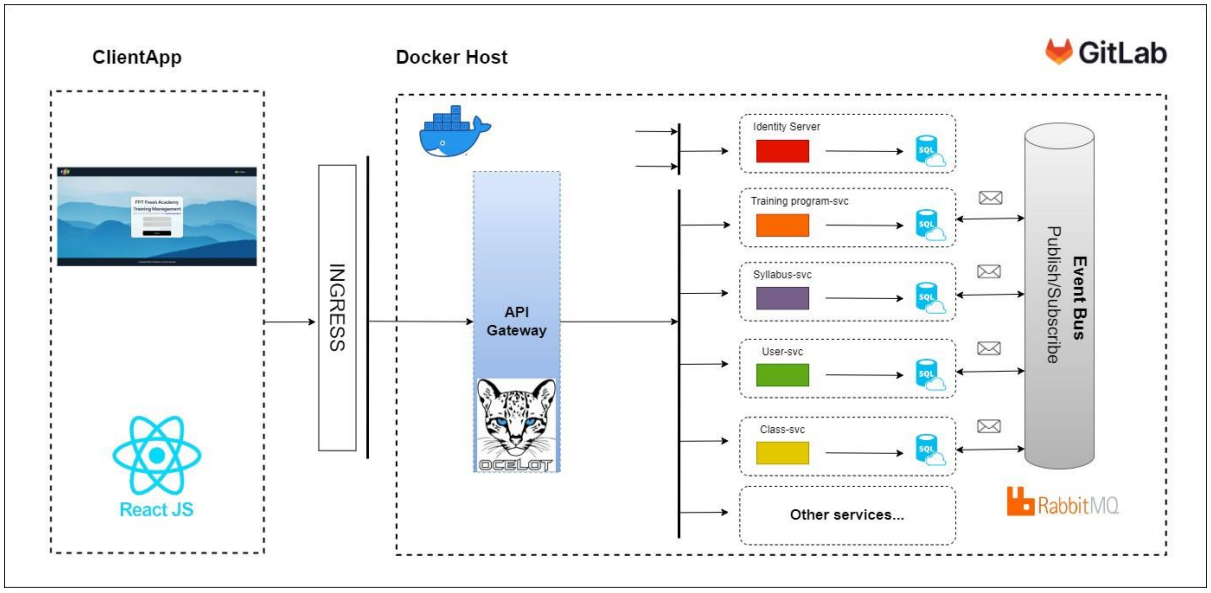


Figure 115 - System architectural design

1.1. Web application architecture description

ASP.NET Web API is a framework developed by Microsoft for building HTTP services that can be consumed by a wide range of clients, including web browsers, mobile devices, and desktop applications. It follows a RESTful architectural style, which emphasizes a client-server communication model using standard HTTP verbs (GET, POST, PUT, DELETE) and resource representations.

The architecture of an ASP.NET Web API application typically consists of the following components:

- Controllers: Controllers are the heart of the Web API application. They handle incoming HTTP requests and generate appropriate responses. Each controller is responsible for a specific set of related actions or resources.
- Models: Models represent the data structures used in the Web API application. They define the shape and properties of the data being transmitted between the client and the server.
- Routes: Routes define the URLs that the Web API application listens to and map them to specific controller actions. They determine how the incoming requests are routed to the appropriate controller.

- HTTP Verbs: HTTP verbs (GET, POST, PUT, DELETE, etc.) are used to perform different actions on the resources exposed by the API. Each HTTP verb is associated with a specific controller action.
- Request Pipeline: The request pipeline is a series of middleware components through which an incoming request passes before reaching the appropriate controller action. These middleware components can perform various tasks such as authentication, authorization, logging, and exception handling.
- Content Negotiation: Web API supports content negotiation, which allows clients to request data in different formats (JSON, XML, etc.). The server can determine the appropriate format based on the client's request headers.
- Filters: Filters are used to add cross-cutting concerns to the Web API application. They can be used for tasks such as authentication, authorization, logging, exception handling, and caching. Filters can be applied globally or at the controller or action level.
- Formatters: Formatters are responsible for serializing and deserializing data between the client and the server. Web API provides built-in formatters for handling common formats like JSON and XML. Custom formatters can also be created to support other formats.
- Dependency Injection: ASP.NET Web API supports dependency injection, allowing you to easily manage and inject dependencies into your controllers and other components. This promotes loose coupling and makes your code more testable and maintainable.

2. Design Overview

- This document describes the technical and user interface design of UAC System. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.
- The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
- The detailed design describes static and dynamic structure for each component and functions. It includes class diagram, class explanations and sequence diagrams for each use cases.
- The database design describes the relationships between entities and details of each entity.
- Document overview:
  - Section 2: gives an overall description of the system architecture design.
  - Section 3: gives component diagrams that describes the connection and integration of the system.
  - Section 4: gives the detail design description which includes class diagram, class explanation and sequence diagram to details the application functions.
  - Section 5: describes screen design.

- Section 6: describes a fully attributed ERD.
- Section 7: describes algorithms.

3. Component Diagram

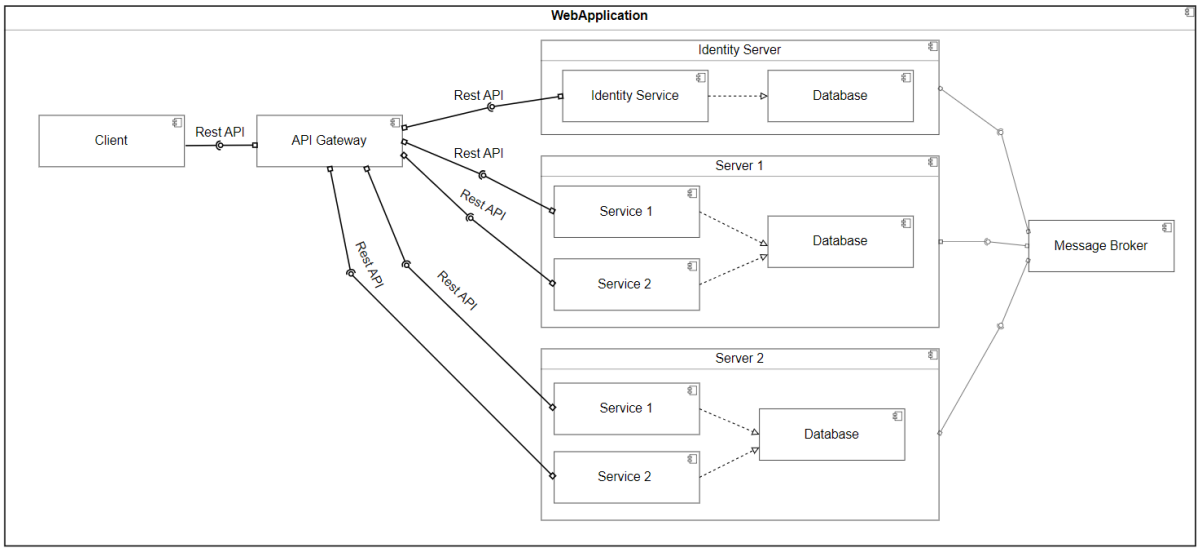


Figure 116 - Component Diagram

Components Dictionary: Describes components	
Microservice	Microservice: A small, independently deployable and scalable software component that implements a specific business capability or function. Microservices communicate with each other through APIs and work together to build a larger application.
API Gateway	API Gateway: A component that acts as a single entry point for clients to access the microservices. It handles requests, performs authentication and authorization, and may provide additional features like caching, rate limiting, and request/response transformations. Ocelot is an example of a tool that can function as an API Gateway in this context.
Message Broker	Message Broker: A communication middleware that enables asynchronous communication between microservices. It allows decoupling of services by sending and receiving messages through channels or topics. Popular message broker implementations is RabbitMQ.
Database	Database: Microservices often have their own dedicated databases to store and manage their data. Depending on the requirements. But in this project, only SQL Server is used.
Authentication and	Authentication and Authorization: Microservices often

Authorization	require mechanisms to authenticate and authorize clients and users. Components like identity providers, JSON Web Tokens (JWT) can be used to handle authentication and authorization processes.
Monitoring and Logging	Monitoring and Logging: Components for monitoring and logging facilitate the collection and analysis of system and application metrics. Elasticsearch are tools that can be used to monitor microservices, gather logs, and perform troubleshooting.
Containerization	Containerization: Microservices are often deployed using containerization technologies like Docker. Containers provide a lightweight and isolated runtime environment for microservices, enabling easy deployment, scaling, and management.
Cloud	A cloud service for storing and managing data for your project, Firebase by Google could be a suitable choice. Firebase provides a comprehensive cloud platform for developing mobile and web applications, including a real-time database and various features such as user authentication, file storage, and analytics.

Table 10 - Components Dictionary

4. Detailed Description

4.1. Class diagram

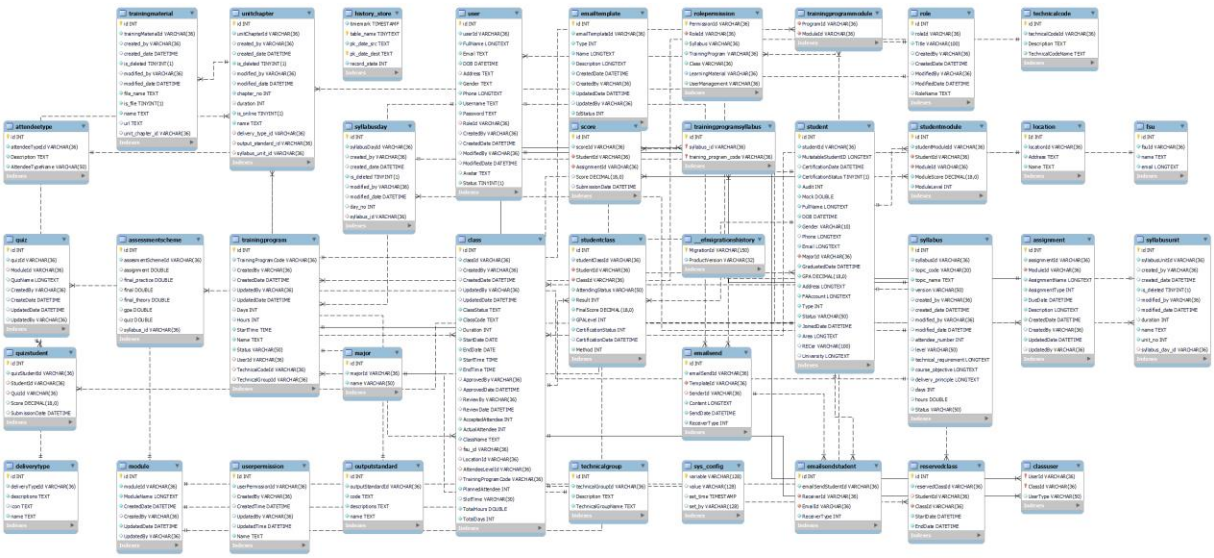


Figure 117 - Class diagram

Class Diagram Dictionary: describe classes	
Class Name	**Description**
AssessmentScheme	Contain information about assessment schemes. It may include fields such as the scheme name, description, evaluation rules, and other related parameters.
Assignment	Contain information about assignments given to students or other users. It may include fields such as the title, description, due date, assigned date, and other relevant assignment details.
AttendeeType	Contain a list of attendee types for events or classes
Class	Contain information about classes. It may include fields such as the class name, description, assigned instructor, start date, end date, and other relevant class details.
ClassUser	Contain information about the association between classes and users. It may include fields such as class ID and user ID to determine user participation in a class.
DeliveryType	Contain different delivery methods or shipping options. Examples include on-site delivery, email delivery, postal delivery, etc.
EmailSend	Contain information about sent emails. It may include fields such as the subject, content, sender, recipient, send date, and other relevant email details.
EmailSendStudent	Contain information about sent emails specifically for students. It may include fields such as student ID, email ID, and other relevant details related to sending emails to students.
EmailTemplate	Contain fields such as the subject, content, and other information required to generate automated emails.
fsu	The exact description of this table is not provided in the question. To understand its functionality, more detailed information about it is required
Location	This table contains information about locations such as the name, address, coordinates, and other relevant details.
Major	This table stores information about academic majors or

	fields of study, including the major name, description, and other related details.
Module	This table represents modules or units of study within a curriculum. It includes fields such as the module name, description, associated resources, and other relevant details.
OutputStandard	This table stores information about output standards or learning outcomes. It includes fields such as the standard name, description, associated competencies, and other details related to the learning outcomes.
Quiz	This table contains information about quizzes or assessments, including the quiz title, description, associated module, duration, and other relevant quiz details.
QuizStudent	This table maintains information about quizzes completed by students, including the student ID, quiz ID, score, and other details related to students' quiz performance.
ReservedClass	This table stores information about reserved or booked classes, including the class ID, reserved date, time slot, and other relevant details.
Role	This table represents different roles or user types within a system, such as student, instructor, administrator, etc.
RolePermission	This table defines the permissions associated with each role, including the role ID, permission ID, and other details related to role-based permissions.
Score	This table stores scores or grades for assessments or assignments, including fields such as student ID, assessment ID, score value, and other relevant score information.
Student	This table contains information about students, including fields such as student ID, name, contact details, and other student-related information.
StudentClass	This table represents the association between students and classes, including fields such as student ID and class ID

	to determine student enrollment in specific classes.
StudentModule	This table stores information about the enrollment of students in specific modules, including fields such as student ID, module ID, enrollment date, and other relevant details.
Syllabus	This table contains information about the syllabus or curriculum, including fields such as syllabus name, description, associated modules, and other relevant syllabus details.
SyllabusDay	This table represents the daily breakdown or schedule within a syllabus, including fields such as syllabus ID, day number, activities, and other details related to the syllabus schedule.
SyllabusTrainingProgram	This table stores information about the training programs associated with a syllabus, including fields such as syllabus ID, program name, description, and other relevant training program details.
SyllabusUnit	This table represents individual units or topics within a syllabus, including fields such as syllabus ID, unit name, description, and other details related to the syllabus units.
TechnicalCode	This table stores technical codes or identifiers, including fields such as code value, description, and other relevant code details.
TechnicalGroup	This table represents technical groups or categories, including fields such as group name, description, and other details related to the technical grouping.
TrainingMaterial	This table stores information about training materials used in a training program. It may include fields such as material ID, title, description, associated modules or syllabus, and other relevant details.
TrainingProgram	This table contains information about training programs or courses. It may include fields such as program ID, title, description, duration, associated modules, and other relevant program details.
TrainingProgramModule	This table represents the relationship between training programs and modules. It may include fields such as

	program ID, module ID, and other details related to the association between programs and modules.
TrainingProgramSyllabus	This table stores information about the mapping between training programs and syllabi. It may include fields such as program ID, syllabus ID, and other relevant details.
UnitChapter	This table represents chapters or sections within a unit. It may include fields such as unit ID, chapter title, description, and other details related to the unit's chapters.
User	This table contains information about users of the system. It may include fields such as user ID, username, password, email, and other user-related details.
UserPermission	This table defines the permissions associated with each user. It may include fields such as user ID, permission ID, and other details related to user permissions.

Table 11 - Components Dictionary

4.2. Class diagram explanation

4.2.1. AssignmentSchemes

Attributes

Attribute	Type	Visibility	Description
id	int	Private	Unique Id of Assignment Scheme
assesmentSchemeId	nvarchar(36)	Private	Another Id of Assignment Scheme
assignment	float	Private	Name of Assignment
final_practice	float	Private	the practical or hands-on component of the final assessment within the scheme.
final	float	Private	the final assessment component within the scheme, which may include a comprehensive exam, project, or other culminating activities.
final_theory	float	Private	This field specifically denotes the theoretical component of the

			final assessment within the scheme.
gpa	float	Private	This field represents the Grade Point Average (GPA) associated with the assignment scheme, which calculates the average grade based on the performance in different assignments and assessments.
quiz	float	Private	This field refers to the quiz component within the scheme, which typically includes short tests or assessments conducted during the course.
syllabus_id	nvarchar(36)	Private	This field indicates the syllabus ID associated with the assignment scheme, linking it to a specific syllabus or curriculum.

4.2.2. Assignment

Attributes

Attribute	Type	Visibility	Description
id	int	Private	This field represents the unique identifier for each assignment in the table.
assignmentId	nvarchar(36)	Private	This field could be an additional identifier or reference for the assignment, which may be used for internal tracking or integration with other systems.
ModuleId	nvarchar(36)	Private	This field indicates the module or unit of study to which the assignment is associated. It links the assignment to a specific module within the curriculum.
AssignmentName	nvarchar(MAX)	Private	This field stores the name or title of the assignment, providing a descriptive label for easy identification.

AssignmentType	int	Private	This field specifies the type or category of the assignment, such as essay, presentation, group project, or problem set.
DueDate	datetime2(7)	Private	This field denotes the deadline or due date for the assignment, indicating the date and time by which the assignment must be completed and submitted.
Description	nvarchar(MAX)	Private	This field provides a description or instructions for the assignment, outlining the requirements, guidelines, and any additional details for the students.
CreatedDate	datetime2(7)	Private	This field records the date and time when the assignment was created or added to the system.
CreatedBy	nvarchar(36)	Private	This field stores the name or identifier of the user who created the assignment.
UpdatedDate	datetime2(7)	Private	This field indicates the date and time when the assignment was last updated or modified.
UpdatedBy	nvarchar(36)	Private	This field stores the name or identifier of the user who last updated or modified the assignment.

4.2.3. Class

Attributes

Attribute	Type	Visibility	Description
id	int	Private	This is a unique field used to identify each class in the table.
classId	nvarchar(36)	Private	Another field that could be another unique identifier used to identify the class, which may be used for linking to other tables or for internal tracking

			purposes.
CreatedBy	nvarchar(36)	Private	This field stores the name or identifier of the person who created the class.
CreatedDate	datetime	Private	This field records the date and time when the class was created or added to the system.
UpdatedBy	nvarchar(36)	Private	This field stores the name or identifier of the person who last updated or modified the class.
UpdatedDate	datetime	Private	This field indicates the date and time when the class was last updated or modified.
ClassStatus	nvarchar(255)	Private	This field describes the status of the class, for example: "Approved," "Pending Approval," "In Progress," "Completed," and other status values.
ClassCode	nvarchar(255)	Private	This field contains a unique identifier code for the class.
Duration	int	Private	This field indicates the expected duration of the class.
StartDate	date	Private	This field specifies the start date of the class.
EndDate	date	Private	This field specifies the end date of the class.
StartTime	time(7)	Private	This field indicates the start time of the class.
EndTime	time(7)	Private	This field indicates the end time of the class.
ApprovedBy	nvarchar(36)	Private	This field stores the name or identifier of the person who approved the class.
ApprovedDate	datetime	Private	This field indicates the date and time when the class was

			approved.
ReviewBy	nvarchar(36)	Private	This field stores the name or identifier of the person who reviewed the class.
ReviewDate	datetime	Private	This field indicates the date and time when the class was reviewed.
AcceptedAttendee	int	Private	This field stores the number of accepted attendees who registered for the class.
ActualAttendee	int	Private	This field stores the actual number of attendees who participated in the class.
ClassName	nvarchar(255)	Private	This field contains the name or a brief description of the class.
fsu_id	nvarchar(36)	Private	This field could be another field used to identify or link the class to other tables.
LocationId	nvarchar(36)	Private	This field identifies the location of the class and may be linked to another table that contains information about the locations.
AttendeeLevelId	nvarchar(36)	Private	This field identifies the level or category of the attendees for the class.
TrainingProgramCode	nvarchar(36)	Private	This field represents the code or identifier of the training program associated with the class.
PlannedAttendee	int	Private	This field stores the planned number of attendees for the class.
SlotTime	nvarchar(30)	Private	This field represents the time slot or session of the class.

4.2.4. Email Send

Attributes

Attribute	Type	Visibility	Description
id	int	Private	This is a unique field used to identify each email send record in the table.
emailSendId	nvarchar(36)	Private	Another field that could be an additional unique identifier or reference for the email send, which may be used for internal tracking or integration with other systems.
TemplateId	nvarchar(36)	Private	This field represents the identifier or reference to the email template used for the email send. It links the email send to a specific template that defines the structure and content of the email.
SenderId	nvarchar(36)	Private	This field stores the identifier or reference to the sender of the email. It could be a user ID or another identifier used to identify the sender.
[Content]	nvarchar(MAX)	Private	This field represents the content of the email, including the subject, body, and any attachments. The actual content of the email is stored in this field.
SendDate	datetime2(7)	Private	This field indicates the date and time when the email was sent.
ReceiverType	int	Private	This field specifies the type of recipient for the email, such as "Individual," "Group," or any other categorization of recipients.

4.2.5. Module

Attributes

Attribute	Type	Visibility	Description

id	int	Private	This field represents the unique identifier for each module in the table.
moduleId	nvarchar(36)	Private	Another field that could be an additional unique identifier used to identify the module.
ModuleName	nvarchar(MAX)	Private	This field stores the name of the module, providing a descriptive label for easy identification.
CreatedDate	datetime2(7)	Private	This field records the date and time when the module was created or added to the system.
CreatedBy	nvarchar(36)	Private	This field stores the name or identifier of the user who created the module.
UpdatedDate	datetime2(7)	Private	This field indicates the date and time when the module was last updated or modified.
UpdatedBy	nvarchar(36)	Private	This field stores the name or identifier of the user who last updated or modified the module.

4.2.6. Quiz

Attributes

Attribute	Type	Visibility	Description
id	int	Private	This attribute serves as a unique identifier for each quiz entry in the table.
quizId	nvarchar(36)	Private	This attribute could be an additional unique identifier or reference for the quiz, potentially used for further identification or integration purposes.
ModuleId	float	Private	This attribute represents the identifier or reference to the module associated with the quiz. It establishes a link between the

			quiz and the specific module it belongs to.
QuizName	float	Private	This attribute stores the name or title of the quiz, providing a descriptive label for easy identification.
CreatedBy	float	Private	This attribute stores the name or identifier of the user who created the quiz.
CreateDate	float	Private	This attribute records the date and time when the quiz was created or added to the system.
UpdatedDate	float	Private	This attribute indicates the date and time when the quiz was last updated or modified.
UpdatedBy	float	Private	This attribute stores the name or identifier of the user who last updated or modified the quiz.

4.2.7. Role

Attributes

Attribute	Type	Visibility	Description
id	int	Private	This attribute represents the unique identifier for each role entry in the table.
roleId	nvarchar(36)	Private	Another attribute that could serve as an additional unique identifier or reference for the role.
Title	nvarchar(100)	Private	This attribute stores the title or name associated with the role, providing a descriptive label for easy identification.
CreatedBy	nvarchar(36)	Private	This attribute stores the name or identifier of the user who created the role.
CreatedDate	datetime	Private	This attribute records the date and time when the role was

			created or added to the system.
ModifiedBy	nvarchar(36)	Private	This attribute stores the name or identifier of the user who last modified or updated the role.
ModifiedDate	datetime	Private	This attribute indicates the date and time when the role was last modified or updated.
RoleName	nvarchar(255)	Private	This attribute represents the name or label of the role, providing a human-readable identifier for the role.

4.2.8. Score

Attributes

Attribute	Type	Visibility	Description
id	int	Private	This attribute represents the unique identifier for each entry in the table.
scoreId	nvarchar(36)	Private	Another attribute that could serve as an additional unique identifier or reference for the score.
StudentId	nvarchar(36)	Private	This attribute represents the identifier or reference to the student associated with the score. It establishes a link between the score and the specific student.
AssignmentId	nvarchar(36)	Private	This attribute represents the identifier or reference to the assignment for which the score is recorded. It links the score to a specific assignment.
Score	float	Private	This attribute stores the numerical or qualitative score achieved by the student for the assignment.
SubmissionDate	datetime2(7)	Private	This attribute indicates the date and time when the assignment

			was submitted and the score was recorded.

4.2.9. Student

Attributes

Attribute	Type	Visibility	Description
id	int	Private	This attribute represents the unique identifier for each student in the table.
MutatableStudentID	nvarchar(450)	Private	Another attribute that could be an additional unique identifier or reference for the student, potentially used for specific purposes like data manipulation.
studentId	nvarchar(36)	Private	This attribute represents the identifier or reference for the student.
FullName	nvarchar(MAX)	Private	This attribute stores the full name of the student.
DOB	datetime2(7)	Private	This attribute represents the date of birth of the student.
Gender	int	Private	This attribute indicates the gender of the student.
Phone	nvarchar(MAX)	Private	This attribute stores the phone number of the student.
Email	nvarchar(MAX)	Private	This attribute stores the email address of the student.
MajorId	nvarchar(36)	Private	This attribute represents the identifier or reference to the major or field of study for the student.
GraduatedDate	datetime2(7)	Private	This attribute indicates the date when the student graduated.
GPA	float	Private	This attribute stores the grade point average of the student.

Address	nvarchar(MAX)	Private	This attribute represents the address of the student.
FAAccount	nvarchar(MAX)	Private	This attribute stores the financial account information of the student.
Type	int	Private	This attribute categorizes the type of student, such as "full-time," "part-time," or other classification criteria.
Status	bit	Private	This attribute represents the status of the student, such as "active," "inactive," or any other relevant status values.
JoinedDate	datetime2(7)	Private	This attribute indicates the date when the student joined or enrolled in the educational institution.
Area	nvarchar(MAX)	Private	This attribute represents the specific area or specialization of the student's major or field of study.
RECer	nvarchar(100)	Private	This attribute stores information about the student's recommender or reference.
University	nvarchar(MAX)	Private	This attribute represents the university or educational institution where the student is enrolled.
CertificationStatus	bit	Private	This attribute indicates the certification status of the student.
CertificationDate	datetime2(7)	Private	This attribute represents the date when the student was certified.
Audit	int	Private	This attribute stores information related to the student's audit or examination.
Mock	float	Private	This attribute represents the

			student's performance or results in mock exams or practice assessments.

4.2.10. AttendeeType

Attributes

Attribute	Type	Visibility	Description
attendeeTypeId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each attendee type.
id	int	Private	This column likely serves as the primary key or another identifier for the attendee type.
Description	nvarchar(255)	Private	This column provides a description or explanation of the attendee type.
AttendeeTypeName	nvarchar(50)	Private	This column specifies the name or label associated with the attendee type.

4.2.11. ClassUser

Attributes

Attribute	Type	Visibility	Description
UserId	nvarchar(36)	Private	This column represents the unique identifier (ID) of a user.
ClassId	nvarchar(36)	Private	This column represents the unique identifier (ID) of a class.
UserType	nvarchar(50)	Private	This column specifies the type or role of the user in relation to the class.

4.2.12. DeliveryType

Attributes

Attribute	Type	Visibility	Description
deliveryTypeId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each

			delivery type.
id	int	Private	This column likely serves as the primary key or another identifier for the delivery type.
descriptions	nvarchar(255)	Private	This column provides a description or explanation of the delivery type.
icon	nvarchar(255)	Private	This column stores the icon or image associated with the delivery type.
name	nvarchar(255)	Private	This column specifies the name or label of the delivery type.

4.2.13. EmailSendStudent

Attributes

Attribute	Type	Visibility	Description
emailSendStudentId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each email send student record.
id	int	Private	This column likely serves as the primary key or another identifier for the email send student record.
ReceiverId	nvarchar(36)	Private	This column represents the unique identifier (ID) of the receiver (e.g., student) of the email.
EmailId	nvarchar(36)	Private	This column represents the unique identifier (ID) of the email being sent.
ReceiverType	int	Private	This column specifies the type or role of the receiver (e.g., student) in relation to the email.

4.2.14. EmailTemplate

Attributes

Attribute	Type	Visibility	Description
emailTemplateId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each email template.
id	int	Private	This column likely serves as the primary key or another identifier for the email template.
Type	int	Private	This column specifies the type or category of the email template.
Name	nvarchar(MAX)	Private	This column stores the name or title of the email template.
Description	nvarchar(MAX)	Private	This column provides a description or explanation of the email template.
CreatedDate	datetime2(7)	Private	This column stores the date and time when the email template was created.
CreatedBy	nvarchar(36)	Private	This column indicates the creator or author of the email template.
UpdatedDate	datetime2(7)	Private	This column stores the date and time when the email template was last updated.
UpdatedBy	nvarchar(36)	Private	This column indicates the user who last updated or modified the email template.

4.2.15. fsu

Attributes

Attribute	Type	Visibility	Description
id	int	Private	This column represents the unique identifier (ID) for each FSU (File Storage Unit) entry.
fsuId	nvarchar(36)	Private	This column likely serves as the primary key or another

			identifier for the FSU.
name	nvarchar(255)	Private	This column stores the name or label associated with the FSU.

4.2.16. Location

Attributes

Attribute	Type	Visibility	Description
locationId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each location entry.
Id	int	Private	This column likely serves as the primary key or another identifier for the location.
Address	nvarchar(255)	Private	This column stores the address information of the location.
Name	nvarchar(255)	Private	This column stores the name or label associated with the location.

4.2.17. Major

Attributes

Attribute	Type	Visibility	Description
majorId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each major.
id	int	Private	This column likely serves as the primary key or another identifier for the major.
name	nvarchar(50)	Private	This column stores the name or label of the major.

4.2.18. OutputStandard

Attributes

Attribute	Type	Visibility	Description

outputStandardId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each output standard.
id	int	Private	This column likely serves as the primary key or another identifier for the output standard.
code	nvarchar(255)	Private	This column stores the code or identifier associated with the output standard.
descriptions	nvarchar(255)	Private	This column provides a description or explanation of the output standard.
	nvarchar(255)	Private	This column specifies the name or label of the output standard.

4.2.19. QuizStudent

Attributes

Attribute	Type	Visibility	Description
quizStudentId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each quiz student record.
id	int	Private	This column likely serves as the primary key or another identifier for the quiz student record.
StudentId	nvarchar(36)	Private	This column represents the unique identifier (ID) of the student.
QuizId	nvarchar(36)	Private	This column represents the unique identifier (ID) of the quiz.
Score	decimal(18, 0)	Private	This column stores the score or grade achieved by the student in the quiz.
SubmissionDate	datetime2(7)	Private	This column stores the date and time when the student

			submitted the quiz.

4.2.20. ReservedClass

Attributes

Attribute	Type	Visibility	Description
reservedClassId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each reserved class.
id	int	Private	This column likely serves as the primary key or another identifier for the reserved class.
StudentId	nvarchar(36)	Private	This column represents the unique identifier (ID) of the student reserving the class.
ClassId	nvarchar(36)	Private	This column represents the unique identifier (ID) of the class being reserved.
Reason	nvarchar(200)	Private	This column stores the reason or explanation for reserving the class.
StartDate	datetime2(7)	Private	This column stores the start date of the reserved class.
EndDate	datetime2(7)	Private	This column stores the end date of the reserved class.

4.2.21. RolePermission

Attributes

Attribute	Type	Visibility	Description
RoleId	nvarchar(36)	Private	This column represents the unique identifier (ID) of a role.
PermissionId	nvarchar(36)	Private	This column represents the unique identifier (ID) of a permission.

4.2.22. StudentClass

Attributes

Attribute	Type	Visibility	Description
studentClassId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each student-class association.
id	int	Private	This column likely serves as the primary key or another identifier for the student-class record.
StudentId	nvarchar(36)	Private	This column represents the unique identifier (ID) of the student.
ClassId	nvarchar(36)	Private	This column represents the unique identifier (ID) of the class.
AttendingStatus	int	Private	This column stores the attending status of the student in the class
Result	int	Private	This column stores the result or outcome of the student's performance in the class.
FinalScore	decimal(18, 0)	Private	This column stores the final score or grade achieved by the student in the class.
GPALevel	int	Private	This column stores the GPA (Grade Point Average) level of the student based on their performance in the class.
CertificationStatus	int	Private	This column indicates the certification status of the student for the class
CertificationDate	datetime2(7)	Private	This column stores the date when the student received the certification (if applicable).
Method	int	Private	This column stores the method or mode of class delivery

4.2.23. StudentModule

Attributes

Attribute	Type	Visibility	Description
studentModuleId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each student-module association.
id	int	Private	This column likely serves as the primary key or another identifier for the student-module record.
StudentId	nvarchar(36)	Private	This column represents the unique identifier (ID) of the student.
ModuleId	nvarchar(36)	Private	This column represents the unique identifier (ID) of the module.
ModuleScore	decimal(18, 0)	Private	This column stores the score or grade achieved by the student in the module.
ModuleLevel	int	Private	This column represents the level or proficiency achieved by the student in the module.

4.2.24. Syllabus

Attributes

Attribute	Type	Visibility	Description
syllabusId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each syllabus.
id	int	Private	This column likely serves as the primary key or another identifier for the syllabus.
topic_code	nvarchar(20)	Private	This column stores the code or identifier associated with a specific topic in the syllabus.

topic_name	nvarchar(255)	Private	This column stores the name or title of the topic in the syllabus.
version	nvarchar(50)	Private	This column indicates the version or revision number of the syllabus.
created_by	nvarchar(36)	Private	This column stores the name or identifier of the user who created the syllabus.
created_date	datetime	Private	This column stores the date and time when the syllabus was created.
modified_by	nvarchar(36)	Private	This column stores the name or identifier of the user who last modified the syllabus.
modified_date	datetime	Private	This column stores the date and time when the syllabus was last modified.
attendee_number	int	Private	This column represents the number of attendees or participants for the syllabus.
[level]	nvarchar(50)	Private	This column may store information related to the level or difficulty of the syllabus (e.g., beginner, intermediate, advanced).
technical_requirement	nvarchar(MAX)	Private	This column stores any technical requirements or prerequisites for the syllabus.
course_objective	nvarchar(MAX)	Private	This column provides the objectives or goals of the course covered by the syllabus.
delivery_principle	varbinary(MAX)	Private	This column describes the delivery principle or approach used for teaching the syllabus content.
days	int	Private	This column represents the number of days required to cover the syllabus.

hours	float	Private	This column represents the total number of hours of instruction or learning time for the syllabus.

4.2.25. SyllabusDay

Attributes

Attribute	Type	Visibility	Description
syllabusDayId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each syllabus day.
id	int	Private	This column likely serves as the primary key or another identifier for the syllabus day record.
created_by	nvarchar(36)	Private	This column stores the name or identifier of the user who created the syllabus day.
created_date	datetime	Private	This column stores the date and time when the syllabus day was created.
is_deleted	bit	Private	This column indicates whether the syllabus day has been deleted or is still active (e.g., "true" or "false").
modified_by	nvarchar(36)	Private	This column stores the name or identifier of the user who last modified the syllabus day.
modified_date	datetime	Private	This column stores the date and time when the syllabus day was last modified.
day_no	int	Private	This column represents the day number or sequence of the syllabus day within the overall syllabus.
syllabus_id	nvarchar(36)	Private	This column references the syllabusId from the syllabus table, linking the syllabus day to

			its parent syllabus.

4.2.26. SyllabusUnit

Attributes

Attribute	Type	Visibility	Description
syllabusUnitId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each syllabus unit.
id	int	Private	This column likely serves as the primary key or another identifier for the syllabus unit record.
created_by	nvarchar(36)	Private	This column stores the name or identifier of the user who created the syllabus unit.
created_date	datetime	Private	This column stores the date and time when the syllabus unit was created.
is_deleted	bit	Private	This column indicates whether the syllabus unit has been deleted or is still active (e.g., "true" or "false").
modified_by	nvarchar(36)	Private	This column stores the name or identifier of the user who last modified the syllabus unit.
modified_date	datetime	Private	This column stores the date and time when the syllabus unit was last modified.
duration	int	Private	This column represents the duration or length of the syllabus unit (e.g., in minutes, hours, or days).
name	nvarchar(255)	Private	This column stores the name or title of the syllabus unit.
unit_no	int	Private	This column represents the unit number or sequence of the syllabus unit within the syllabus

			day.
syllabus_day_id	nvarchar(36)	Private	This column references the syllabusDayId from the syllabus day table, linking the syllabus unit to its parent syllabus day.

4.2.27. TechnicalCode

Attributes

Attribute	Type	Visibility	Description
technicalCodeId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each technical code.
id	int	Private	This column likely serves as the primary key or another identifier for the technical code.
Description	nvarchar(255)	Private	This column stores the description or details associated with the technical code.
TechnicalCodeName	nvarchar(255)	Private	This column stores the name or label of the technical code.

4.2.28. TechnicalGroup

Attributes

Attribute	Type	Visibility	Description
technicalGroupId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each technical group.
id	int	Private	This column likely serves as the primary key or another identifier for the technical group.
Description	nvarchar(255)	Private	This column stores the description or details associated with the technical group.
TechnicalGroupName	nvarchar(255)	Private	This column stores the name or label of the technical group.

4.2.29. TrainingMaterial

Attributes

Attribute	Type	Visibility	Description
trainingMaterialId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each training material.
id	int	Private	This column likely serves as the primary key or another identifier for the training material.
created_by	nvarchar(36)	Private	This column stores the name or identifier of the user who created the training material.
created_date	datetime	Private	This column stores the date and time when the training material was created.
is_deleted	bit	Private	This column indicates whether the training material has been deleted or is still active
modified_by	nvarchar(36)	Private	This column stores the name or identifier of the user who last modified the training material.
modified_date	datetime	Private	This column stores the date and time when the training material was last modified.
file_name	nvarchar(255)	Private	This column stores the name of the file associated with the training material.
is_file	bit	Private	This column indicates whether the training material is a file or another type of resource
name	nvarchar(255)	Private	This column stores the name or title of the training material.
url	nvarchar(255)	Private	This column stores the URL or web address of the training material if it is not a file.

unit_chapter_id	nvarchar(36)	Private	This column references the unit_chapter_id from another table, linking the training material to a specific unit or chapter.

4.2.30. TrainingProgram

Attributes

Attribute	Type	Visibility	Description
TrainingProgramCode	nvarchar(36)	Private	This column represents the unique identifier (ID) for each training program code.
id	int	Private	This column likely serves as the primary key or another identifier for the training program code.
CreatedBy	nvarchar(36)	Private	This column stores the name or identifier of the user who created the training program code.
CreatedDate	datetime	Private	This column stores the date and time when the training program code was created.
UpdatedBy	nvarchar(36)	Private	This column stores the name or identifier of the user who last updated the training program code.
UpdatedDate	datetime	Private	This column stores the date and time when the training program code was last updated.
Days	int	Private	This column represents the number of days the training program lasts.
Hours	int	Private	This column represents the total number of hours of instruction or learning time for the training program.
StartTime	time(7)	Private	This column stores the starting

			time or schedule for the training program.
Name	nvarchar(255)	Private	This column stores the name or title of the training program.
Status	nvarchar(50)	Private	This column indicates the status of the training program (e.g., active, inactive, completed).
UserId	nvarchar(36)	Private	This column references the ID of the user associated with the training program.
TechnicalCodeId	nvarchar(36)	Private	This column references the technicalCodeId from the technical code table, linking the training program to a specific technical code.
TechnicalGroupId	nvarchar(36)	Private	This column references the technicalGroupId from the technical group table, linking the training program to a specific technical group.

4.2.31. TrainingProgramModule

Attributes

Attribute	Type	Visibility	Description
ProgramId	nvarchar(36)	Private	This column represents the identifier (ID) or reference to the training program associated with the module.
ModuleId	nvarchar(36)	Private	This column represents the identifier (ID) or reference to the module included in the training program.

4.2.32. UnitChapter

Attributes

Attribute	Type	Visibility	Description
unitChapterId	nvarchar(36)	Private	This column represents the

			unique identifier (ID) for each unit chapter.
id	int	Private	This column likely serves as the primary key or another identifier for the unit chapter.
created_by	nvarchar(36)	Private	This column stores the name or identifier of the user who created the unit chapter.
created_date	datetime	Private	This column stores the date and time when the unit chapter was created.
is_deleted	bit	Private	This column indicates whether the unit chapter has been deleted or is still active
modified_by	nvarchar(36)	Private	This column stores the name or identifier of the user who last modified the unit chapter.
modified_date	datetime	Private	This column stores the date and time when the unit chapter was last modified.
chapter_no	int	Private	This column represents the number or sequence of the chapter within the unit.
duration	int	Private	This column represents the duration or length of the unit chapter
is_online	bit	Private	This column indicates whether the unit chapter is available online or delivered through online means
name	nvarchar(255)	Private	This column stores the name or title of the unit chapter.
delivery_type_id	nvarchar(36)	Private	This column references the delivery_type_id from another table, linking the unit chapter to a specific delivery type.
output_standard_id	nvarchar(36)	Private	This column references the output_standard_id from

			another table, linking the unit chapter to a specific output standard.
syllabus_unit_id	nvarchar(36)	Private	This column references the syllabus_unit_id from another table, linking the unit chapter to a specific syllabus unit.

4.2.33. User

Attributes

Attribute	Type	Visibility	Description
userId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each user.
id	int	Private	This column likely serves as the primary key or another identifier for the user.
FullName	nvarchar(MAX)	Private	This column stores the full name or complete name of the user.
Email	nvarchar(255)	Private	This column stores the email address of the user.
DOB	datetime2(7)	Private	This column stores the date of birth of the user.
Address	nvarchar(255)	Private	This column stores the address or physical location of the user.
Gender	nvarchar(255)	Private	This column stores the gender of the user.
Phone	nvarchar(MAX)	Private	This column stores the phone number or contact number of the user.
Username	nvarchar(255)	Private	This column stores the unique username chosen by the user for authentication purposes.

Password	nvarchar(255)	Private	This column stores the encrypted or hashed password of the user for authentication.
RoleId	nvarchar(36)	Private	This column references the ID of the user's role or role group in the system.
CreatedBy	nvarchar(36)	Private	This column stores the name or identifier of the user who created the user account.
CreatedDate	datetime	Private	This column stores the date and time when the user account was created.
ModifiedBy	nvarchar(36)	Private	This column stores the name or identifier of the user who last modified the user account.
ModifiedDate	datetime	Private	This column stores the date and time when the user account was last modified.
Avatar	nvarchar(255)	Private	This column stores the profile picture or avatar of the user.
Status	bit	Private	This column indicates the status of the user account (e.g., active, inactive, suspended).

4.2.34. UnitChapter

Attributes

Attribute	Type	Visibility	Description
userPermissionId	nvarchar(36)	Private	This column represents the unique identifier (ID) for each user permission.
Id	int	Private	This column likely serves as the primary key or another identifier for the user permission.
CreatedBy	nvarchar(36)	Private	column stores the name or identifier of the user who

			created the user permission.
CreatedTime	datetime	Private	This column stores the date and time when the user permission was created.
UpdatedBy	nvarchar(36)	Private	This column stores the name or identifier of the user who last updated the user permission.
UpdatedTime	datetime	Private	This column stores the date and time when the user permission was last updated.
Name	nvarchar(255)	Private	This column stores the name or title of the user permission.

4.3. Interaction diagram

4.3.1. User Management Diagram

4.3.1.1. <Guest> Login

Summary: This diagram show how guest login

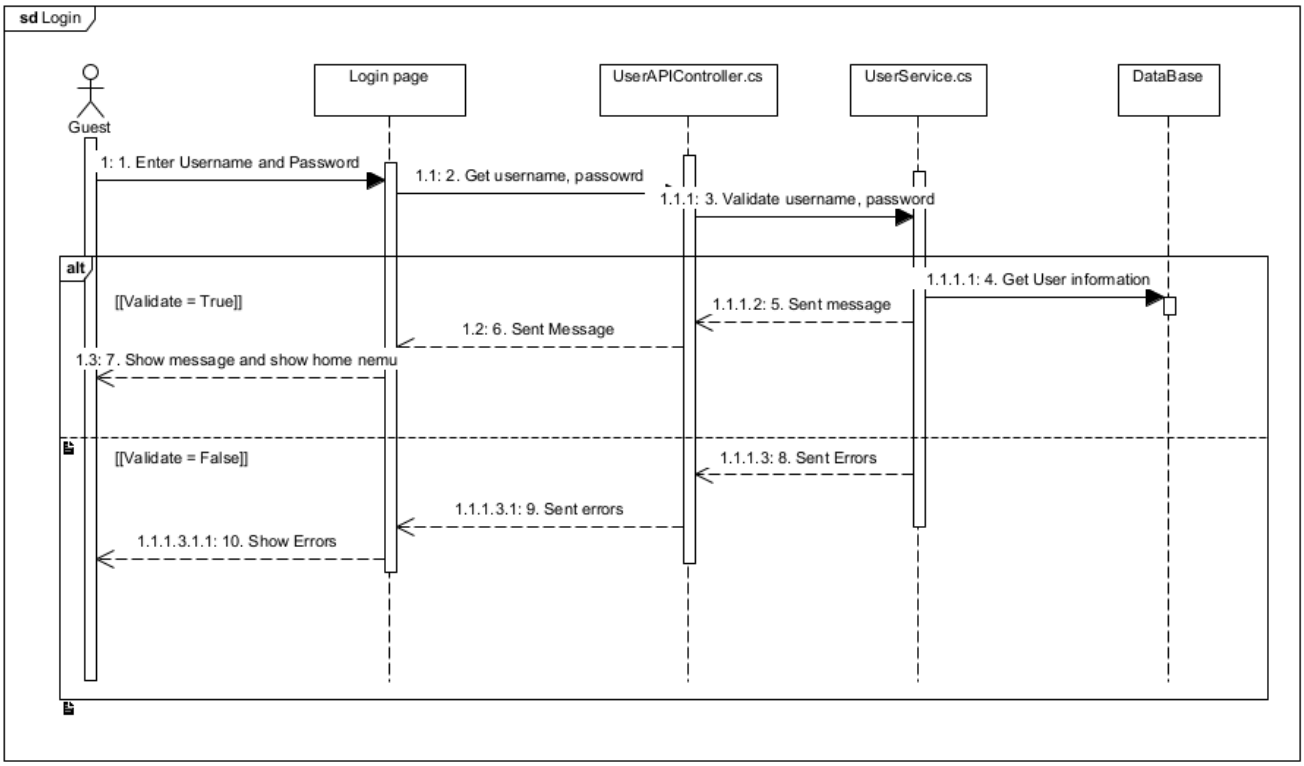


Figure 118 - <Guest> Login

4.3.1.2. <Guest> Forgot Password

Summary: This diagram show how guest forgot password

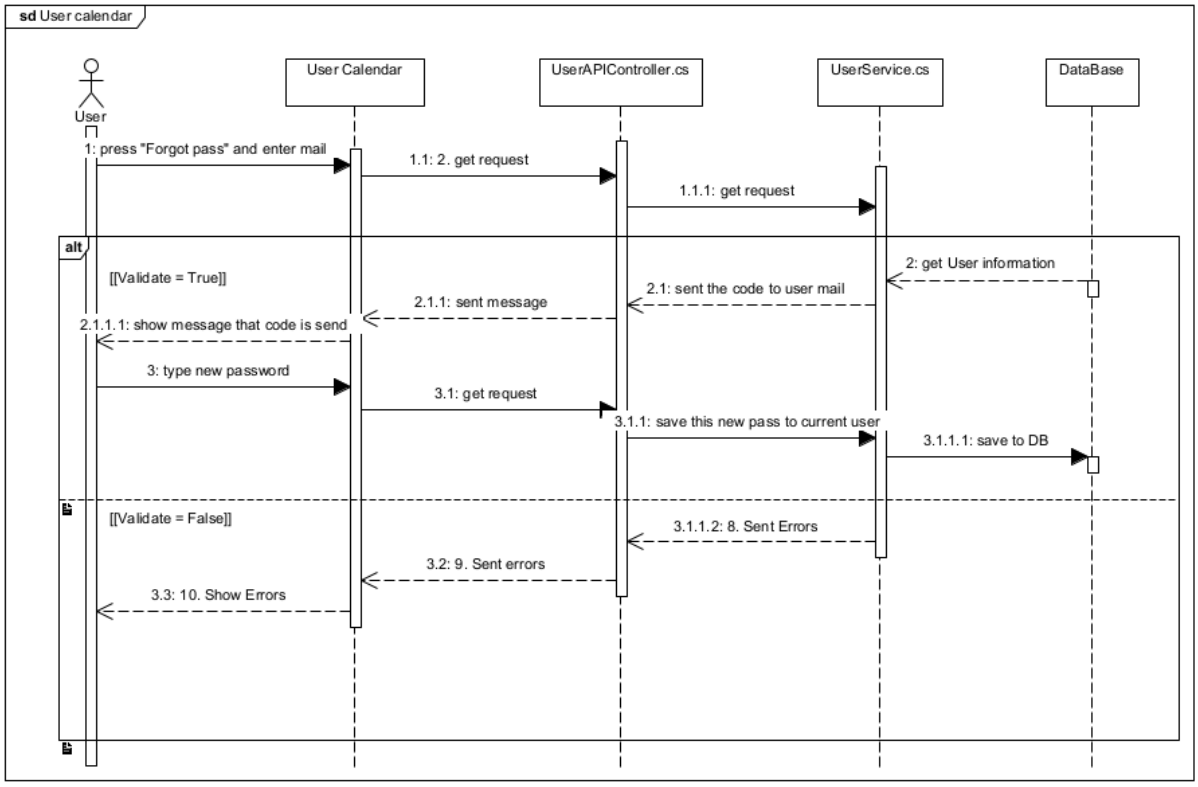


Figure 119 - <Guest> Forgot Password

4.3.1.3. <Super Admin> Edit User

Summary: This diagram show how the Super Admin edit user

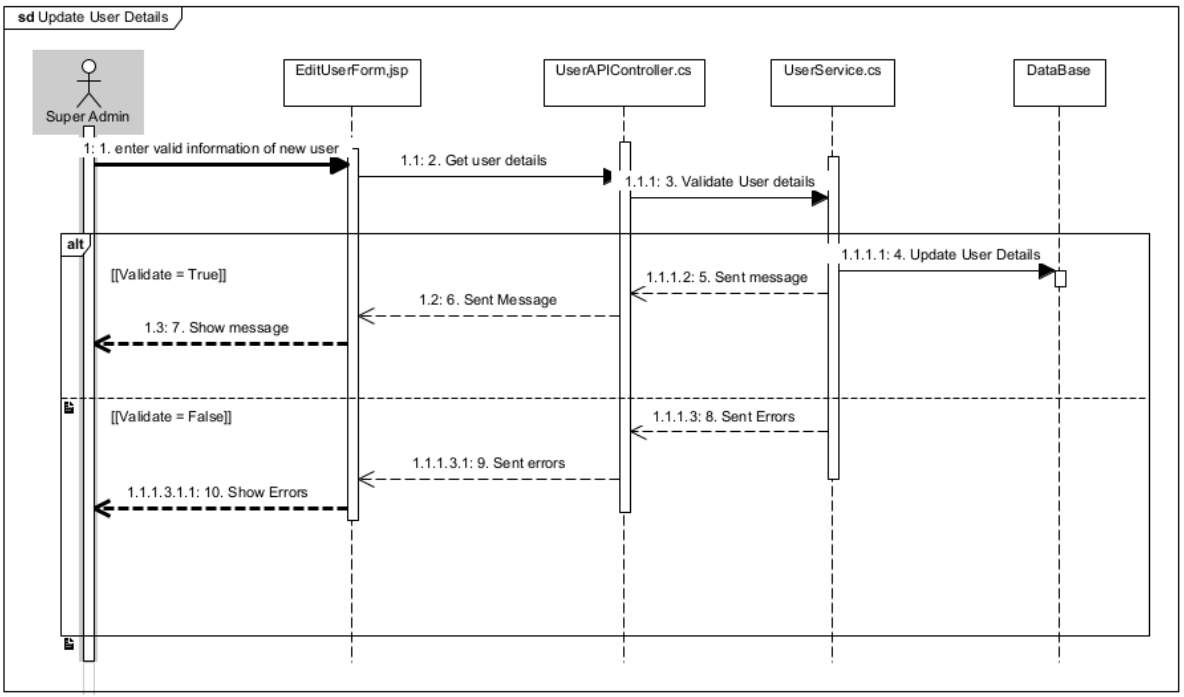


Figure 120 - <Super Admin> Edit User

4.3.1.4. <Super Admin>Create User

Summary: This diagram show how the Super Admin create user

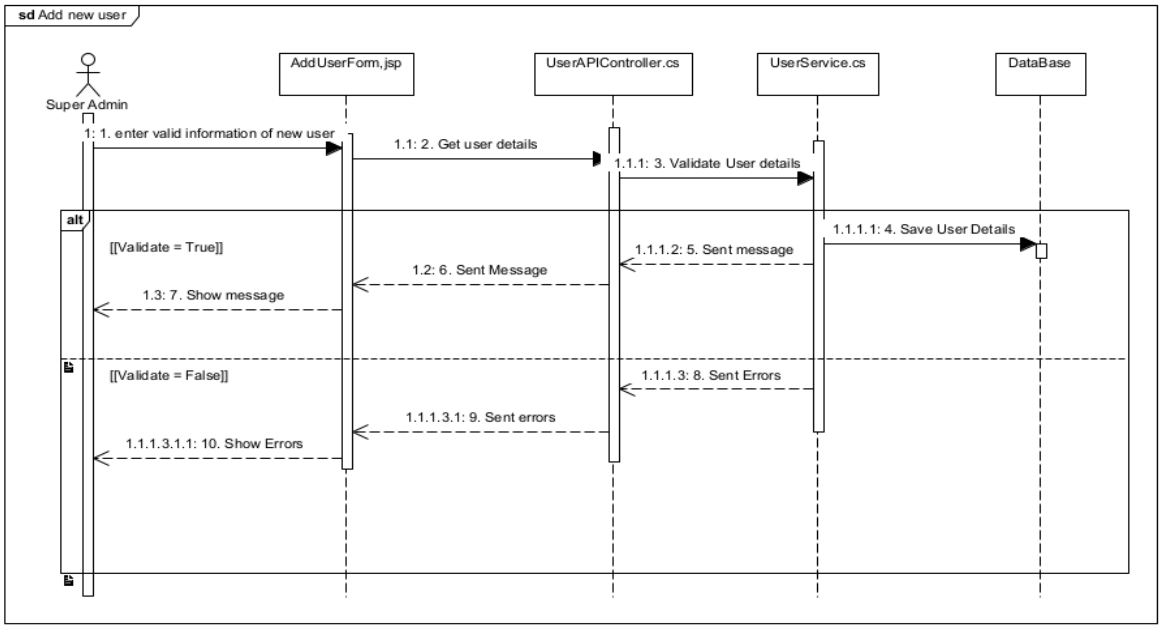


Figure 121 - <Super Admin> Create User

4.3.1.5. <Super Admin> Change User Role

Summary: This diagram show how the Super Admin change user role

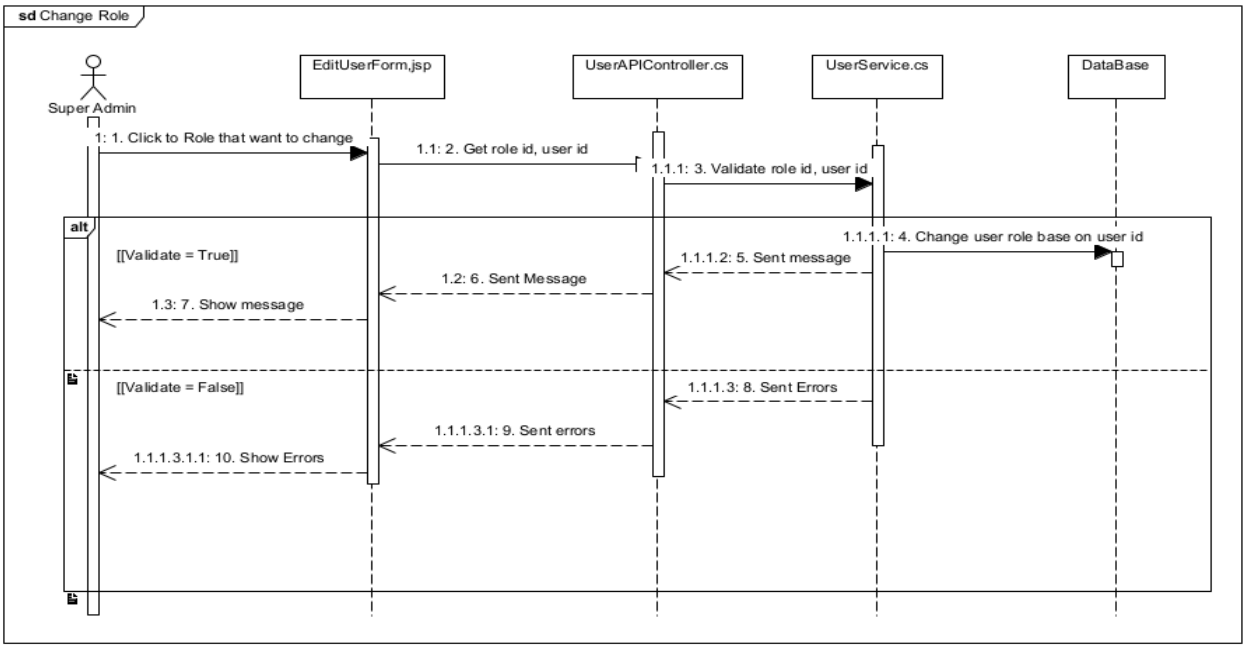


Figure 122 - <Super Admin> Change User Role

4.3.1.6. <Super Admin> Change Status

Summary: This diagram show how the Super Admin change status

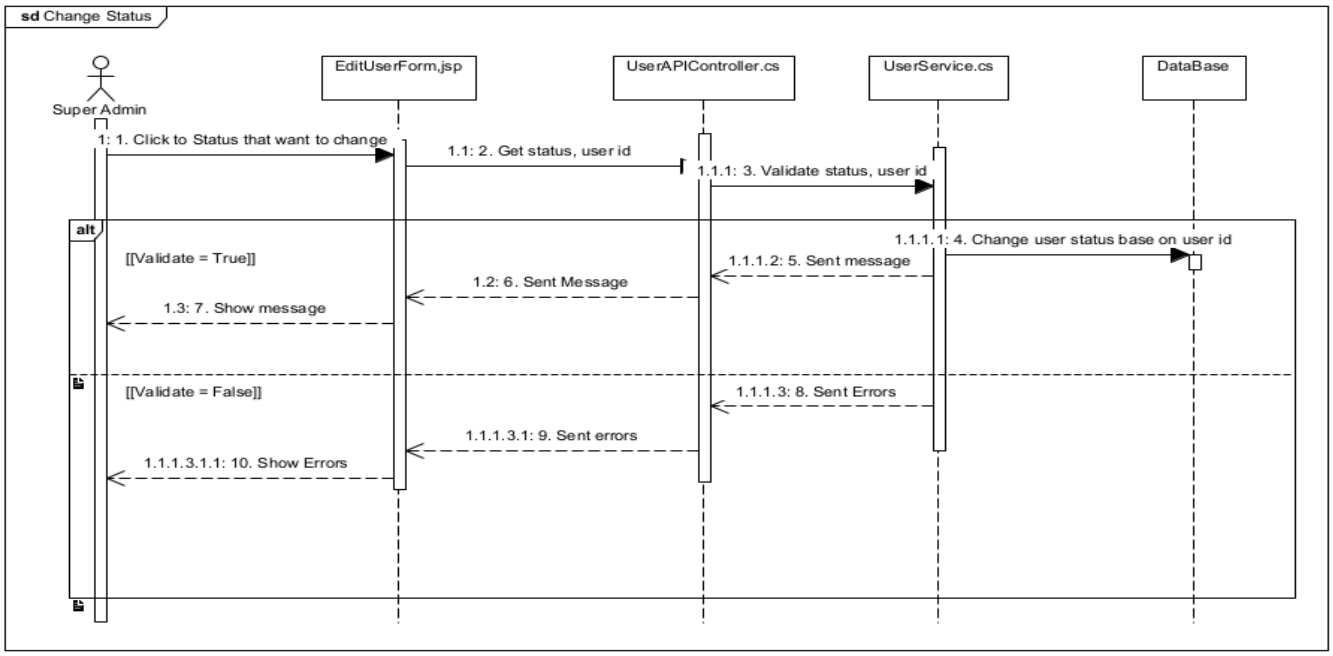


Figure 123 - <Super Admin> Change Status

4.3.1.7. <Super Admin> Soft User List

Summary: This diagram show how the Super Admin edit user

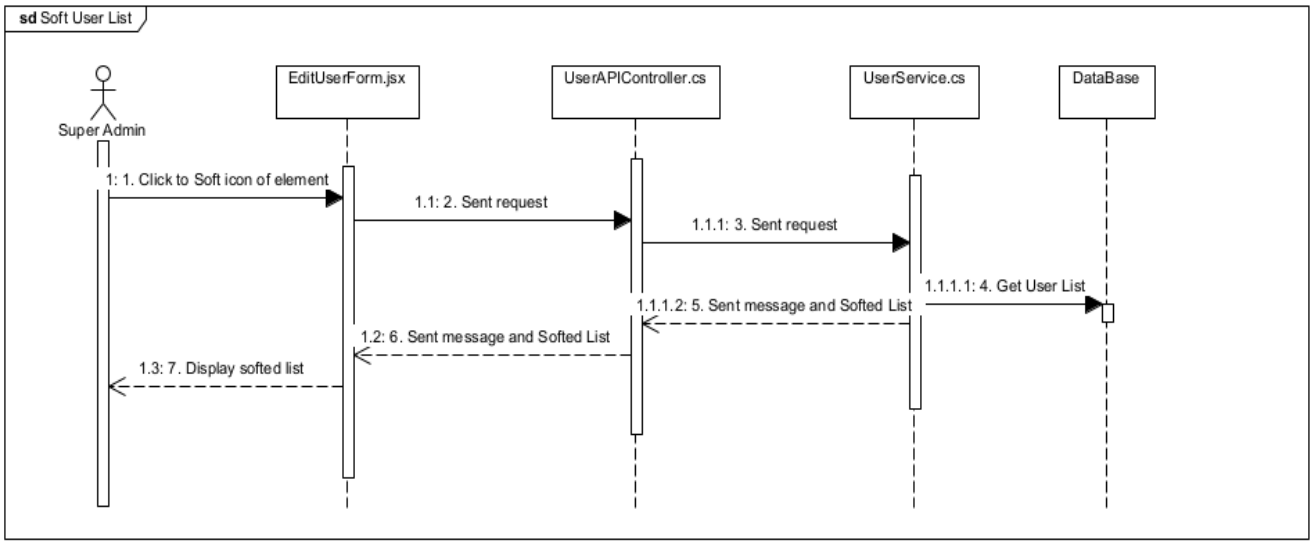


Figure 124 - <Super Admin> Soft User List

4.3.1.8. <Super Admin> Search User

Summary: This diagram show how the Super Admin search user

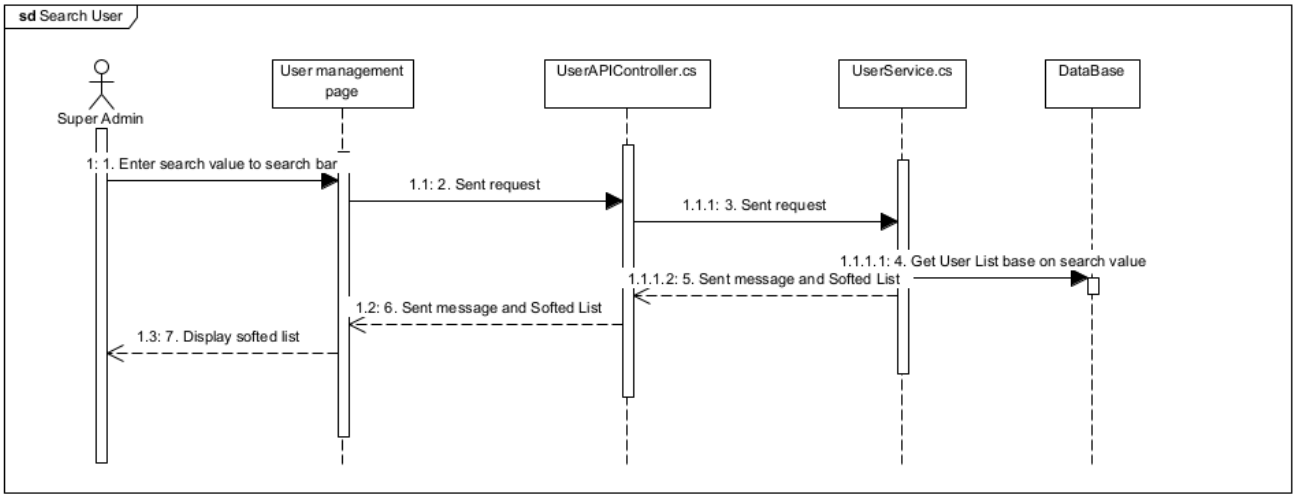


Figure 125 - <Super Admin> Search User

4.3.1.9. <Super Admin> Filter User

Summary: This diagram show how the Super Admin filter user

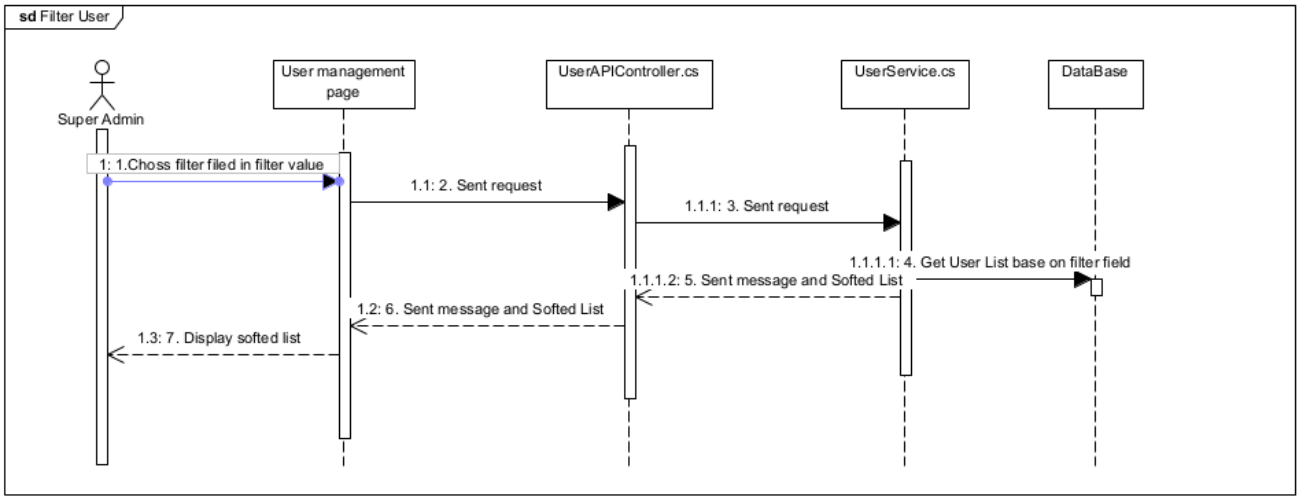


Figure 126 - <Super Admin> Filter User

4.3.1.10. <Super Admin> View User List

Summary: This diagram show how the Super Admin view user list

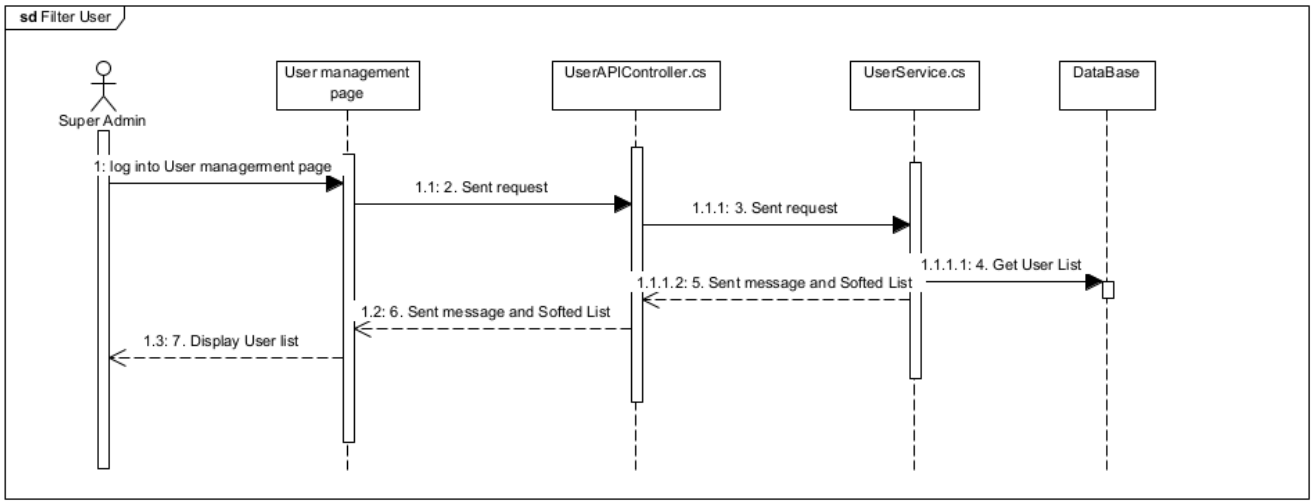


Figure 127 - <Super Admin> View User List

4.3.1.11. <Super Admin> View User's Permission List

Summary: This diagram show how the Super Admin view user's permission list

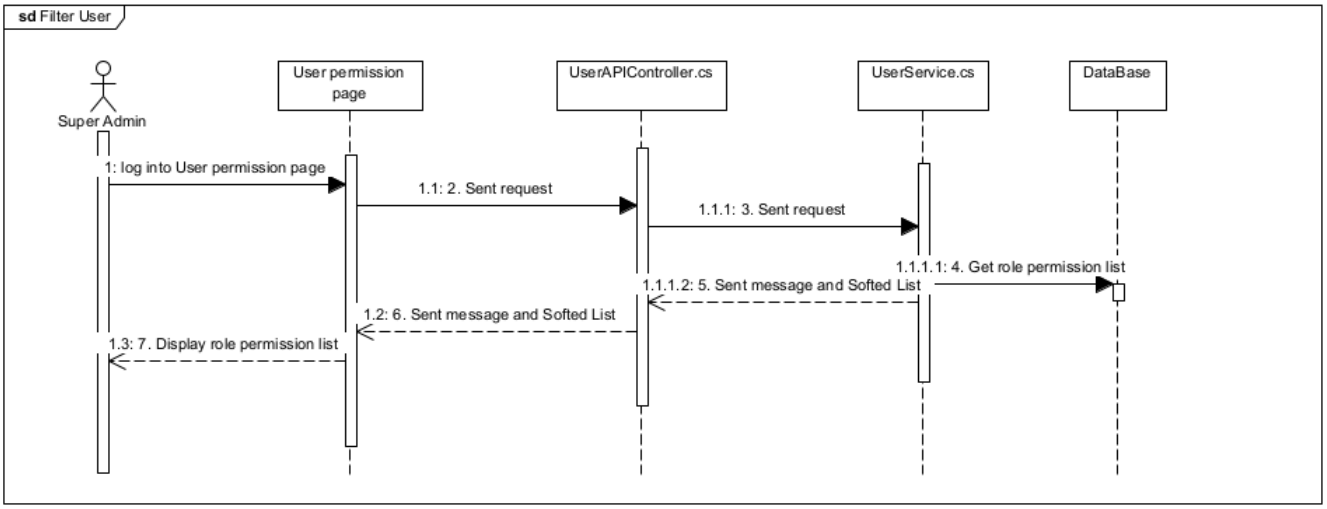


Figure 128 - <Super Admin> Change User Role

4.3.1.12. <Super Admin> Update User's Permission List

Summary: This diagram show how the Super Admin update user's permission list

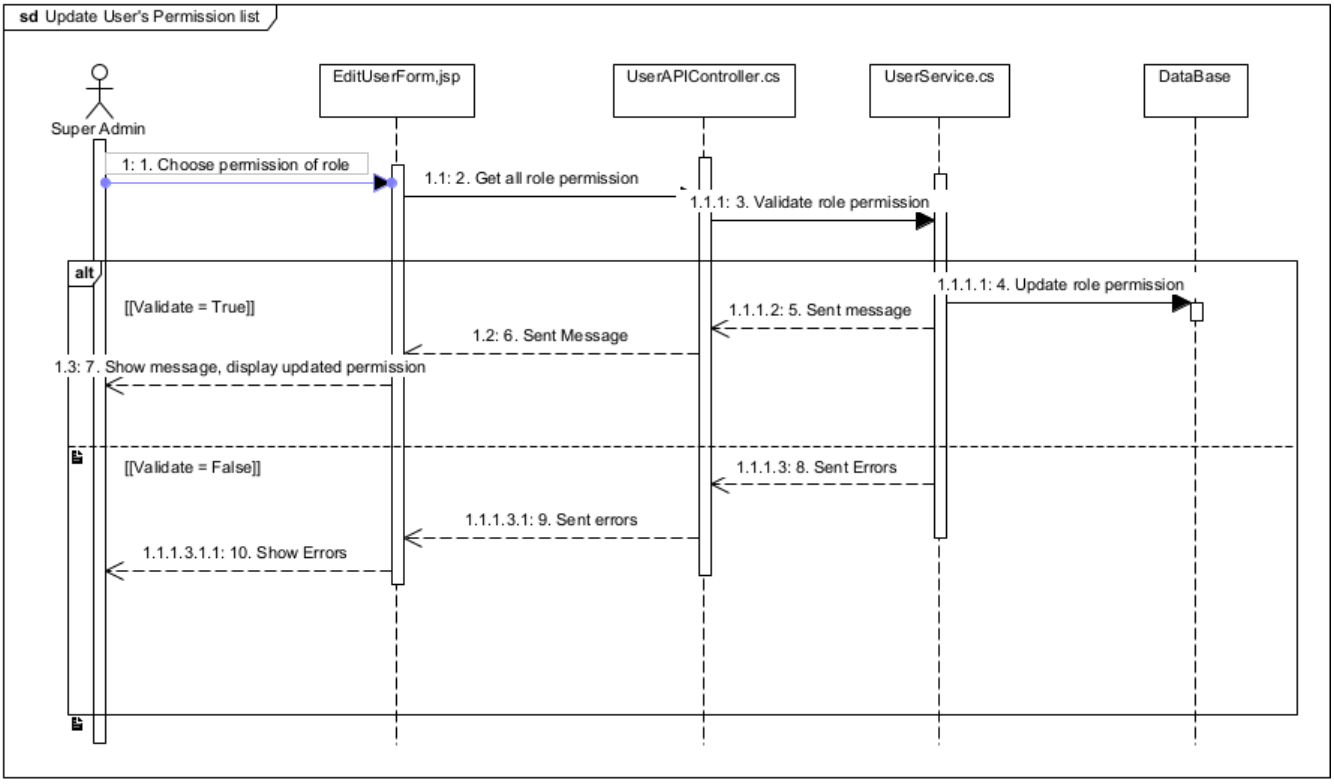


Figure 129 - <Super Admin> Update User's Permission List

4.3.1.13. <Super Admin> Import User List

Summary: This diagram show how the Super Admin import user list

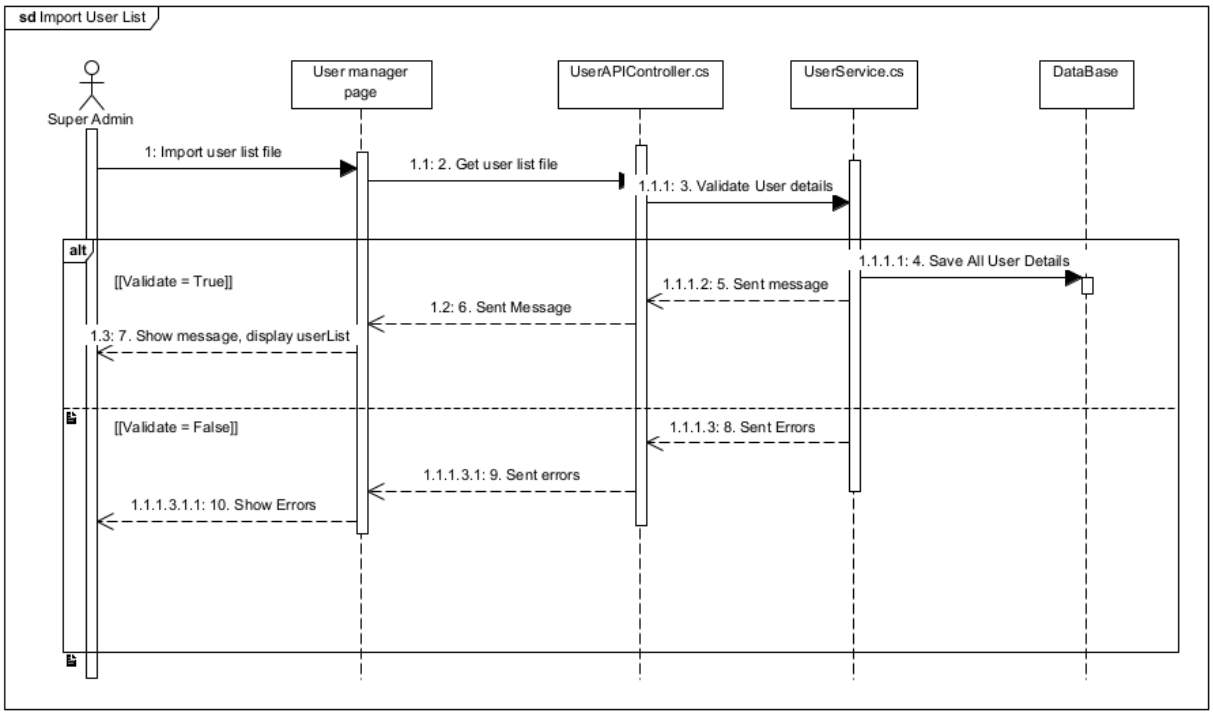


Figure 130 - <Super Admin> Import User List

4.3.1.14. <Super Admin> Import New Role

Summary: This diagram show how the Super Admin import new role

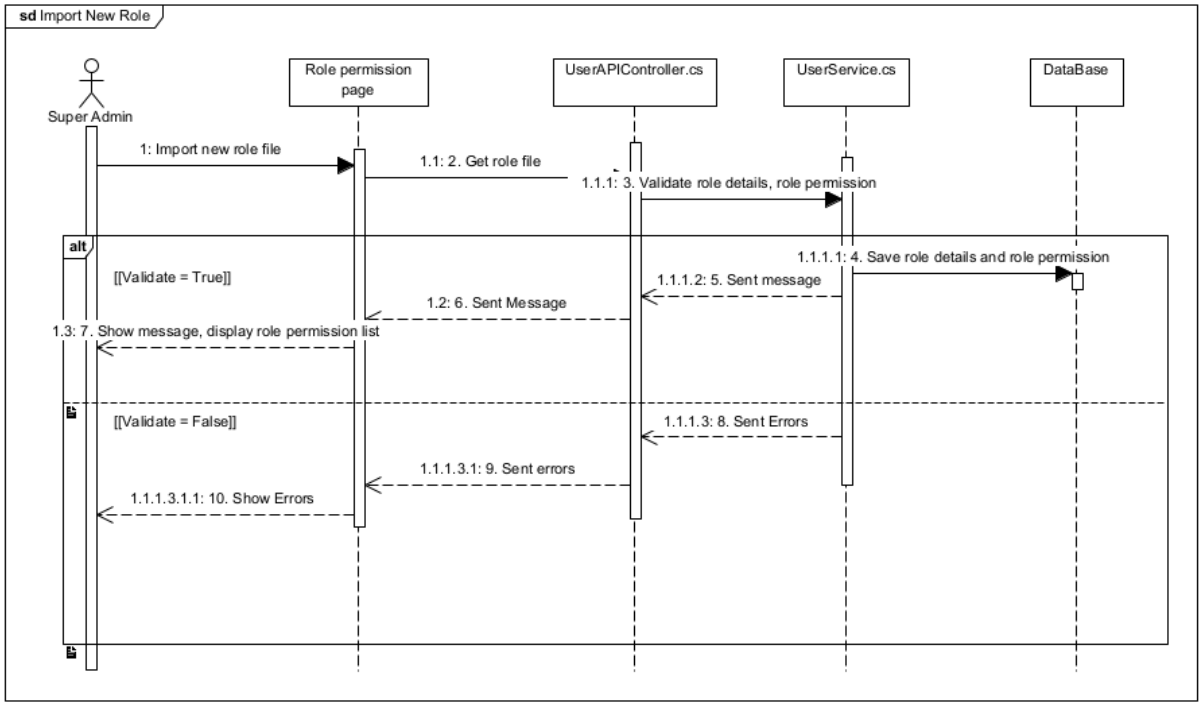


Figure 131 - <Super Admin> Import New Role

4.3.1.15. <Super Admin> Import Role Permission

Summary: This diagram show how the Super Admin import role permission

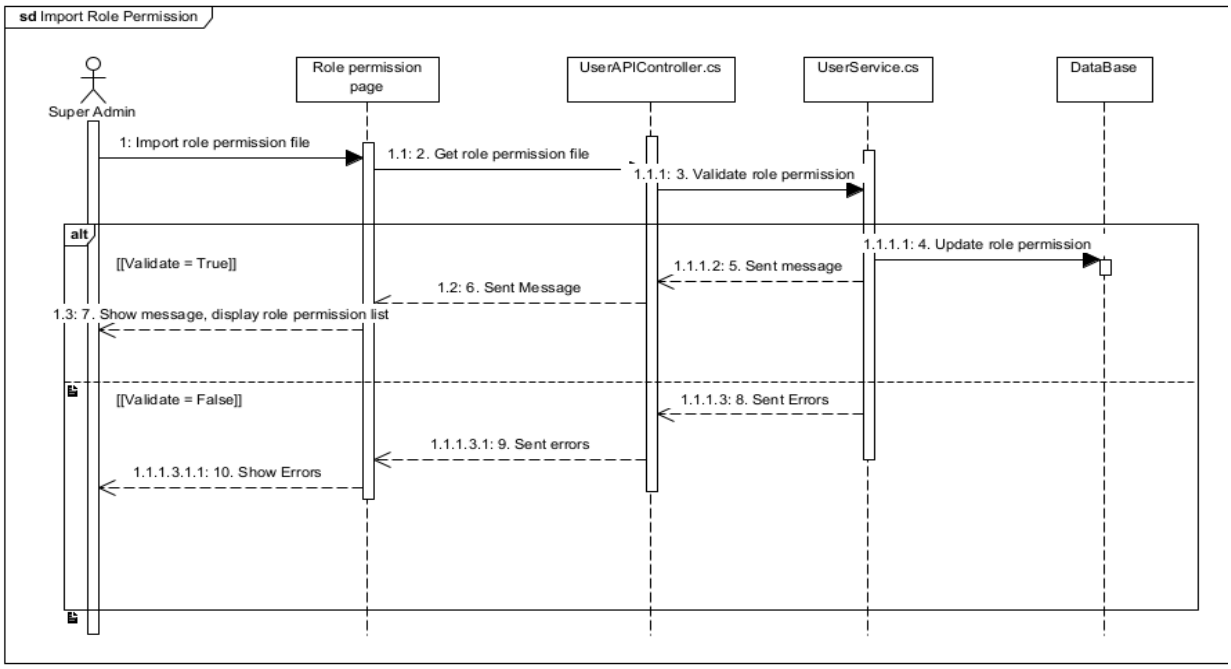


Figure 132 - <Super Admin> Import Role Permission

4.3.1.16. <Super Admin> Export User List

Summary: This diagram show how the Super Admin export user list

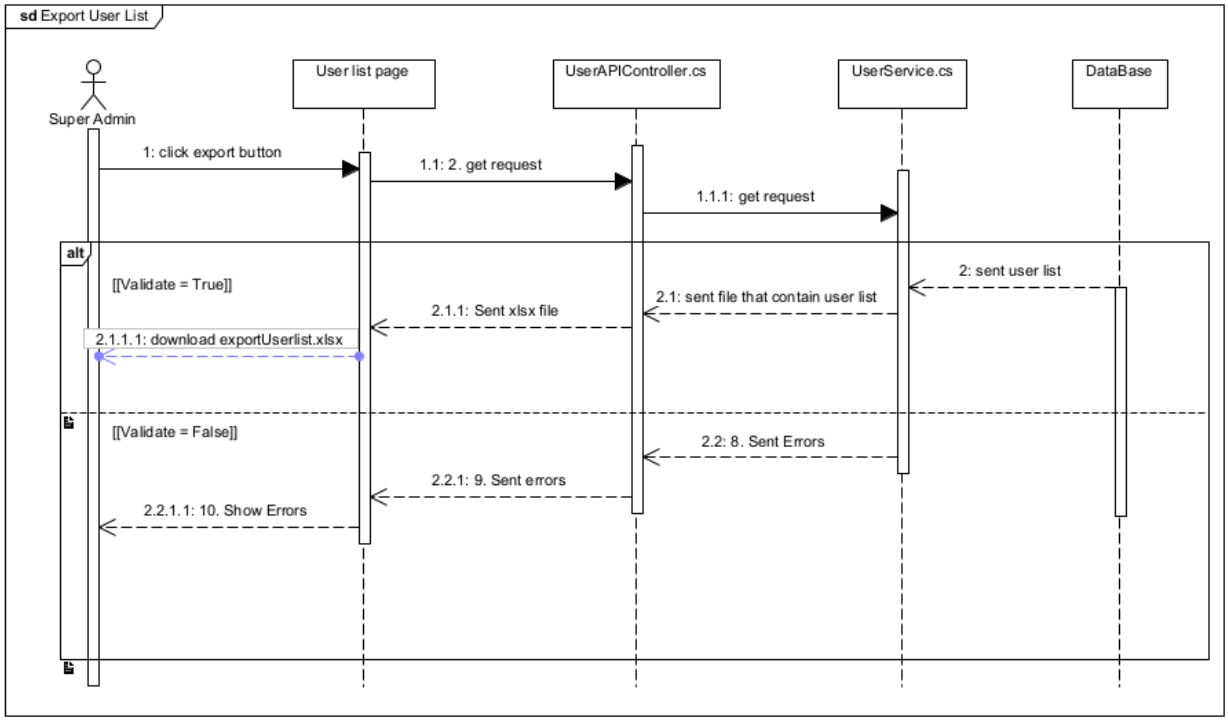


Figure 133 - <Super Admin> Export User List

4.3.1.17. <Super Admin> Export User's Permission List

Summary: This diagram show how the Super Admin export user's permission list

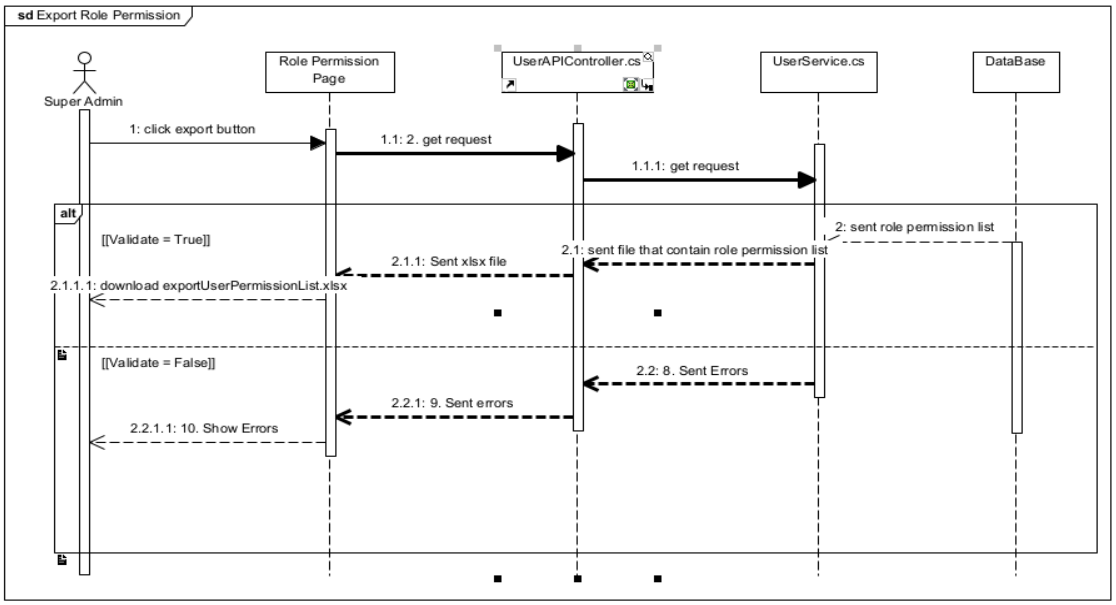


Figure 134 - <Super Admin> Export User's Permission List

4.3.1.18. <User> User Calendar

Summary: This diagram show how the user view user calendar

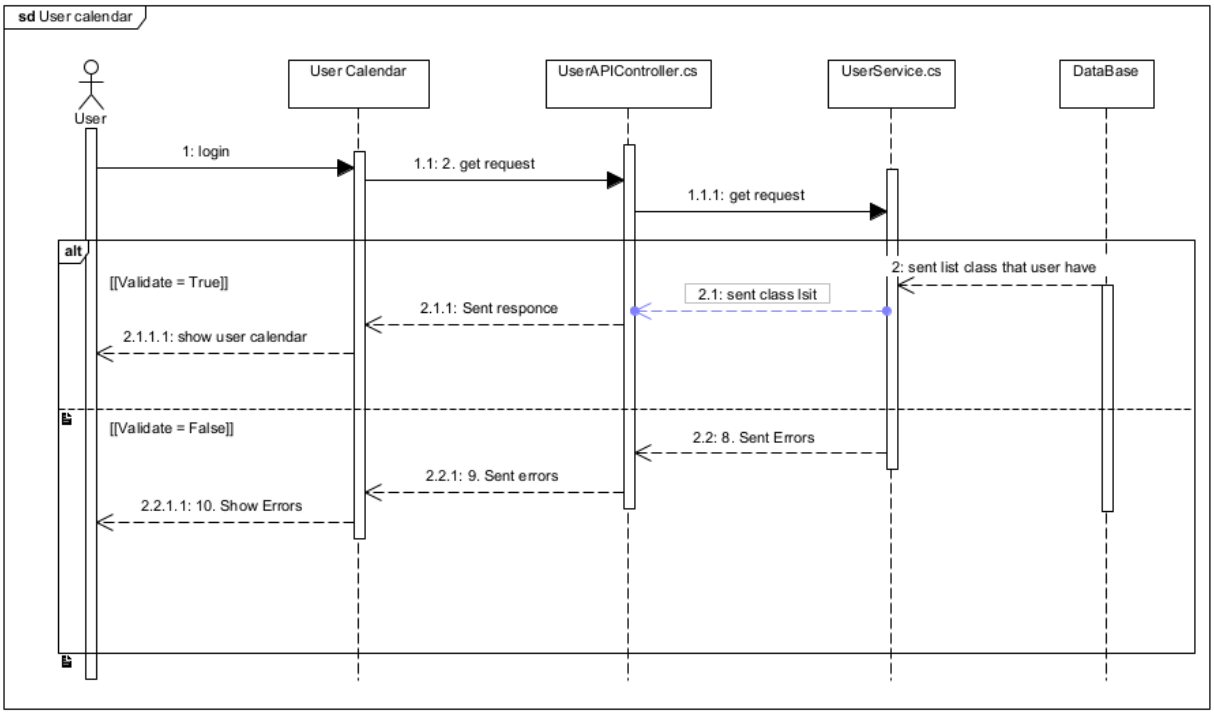


Figure 135 - <User> User Calendar

4.3.2. Syllabus Management Diagram

4.3.2.1. <Super Admin, Class Admin, Trainer> View list Syllabus

Summary: This diagram show how the Super Admin, Class Admin, Trainer view list Syllabus

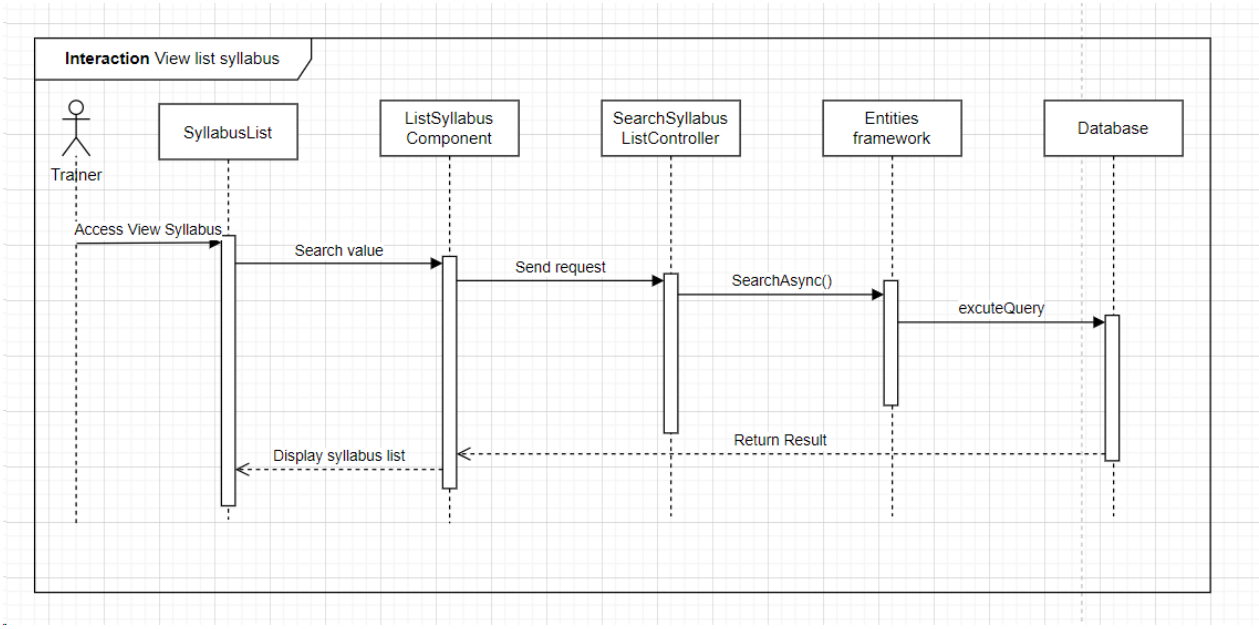


Figure 136 - <Super Admin, Class Admin, Trainer> View list Syllabus

4.3.2.2. <Super Admin, Class Admin, Trainer> Search by Date picker

Summary: This diagram show how the Super admin, Class admin, Trainer search by Date picker

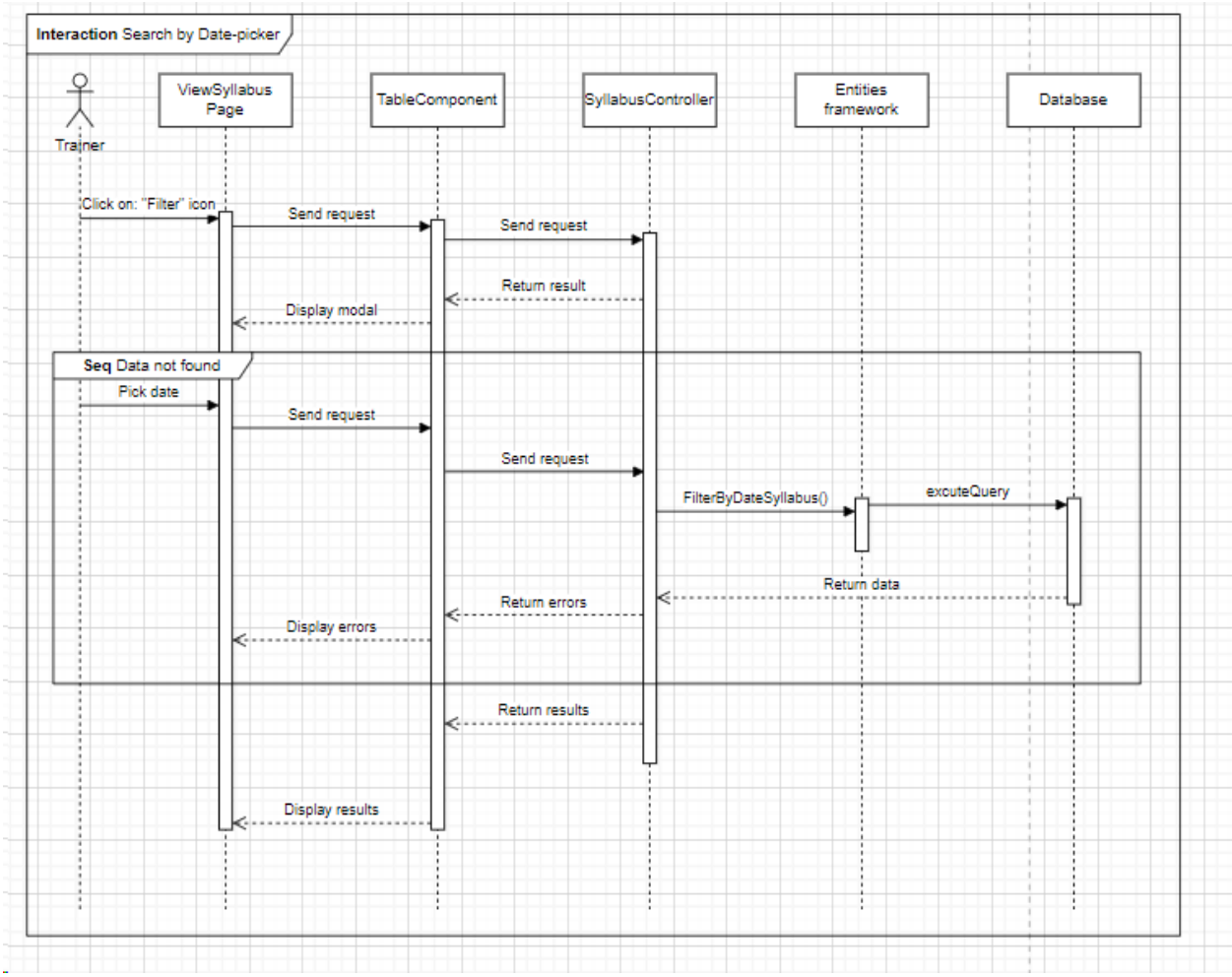


Figure 137 - <Super Admin, Class Admin, Trainer> Search by Date picker

4.3.2.3. <Super Admin, Class Admin> Import syllabus

Summary: This diagram show how the Super Admin, Class Admin import syllabus

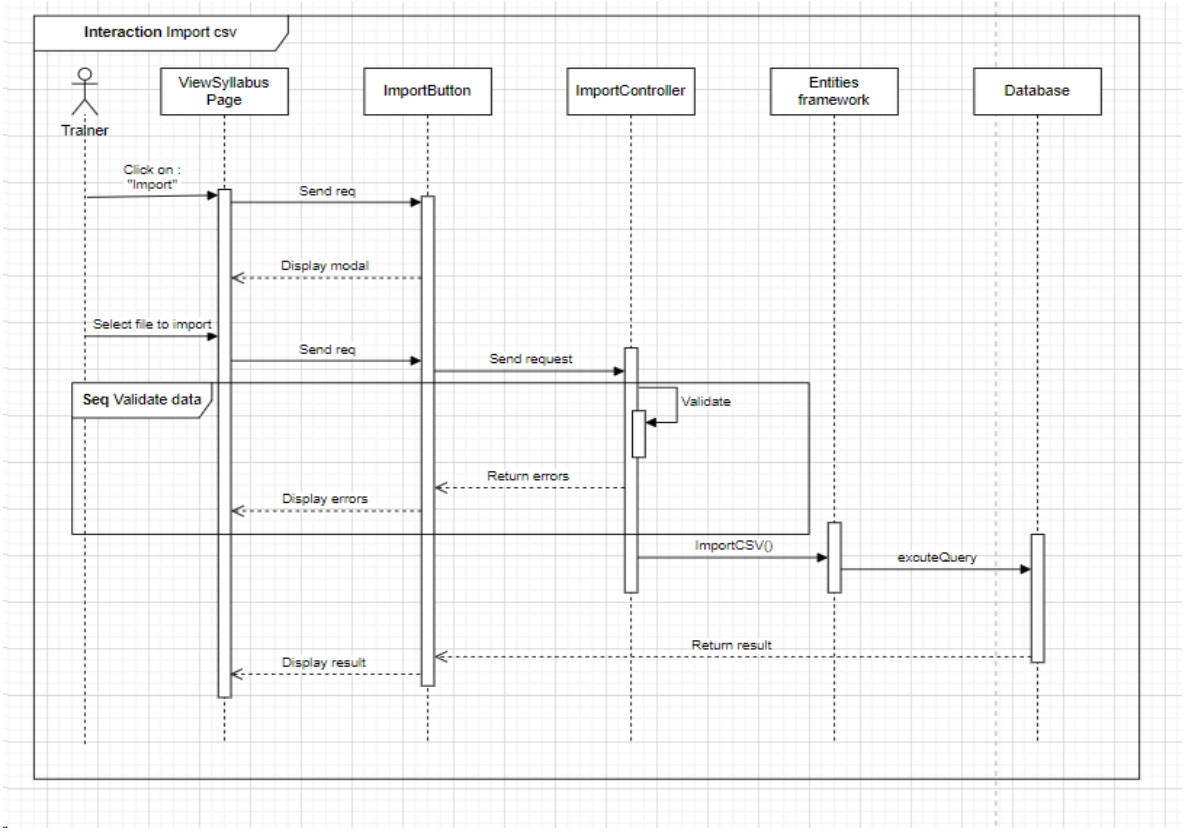


Figure 138 - <Super Admin, Class Admin> Import syllabus

4.3.2.4. <Super Admin, Class Admin, Trainer> Create Syllabus

Summary: This diagram show how the Super Admin, Class Admin, Trainer create Syllabus

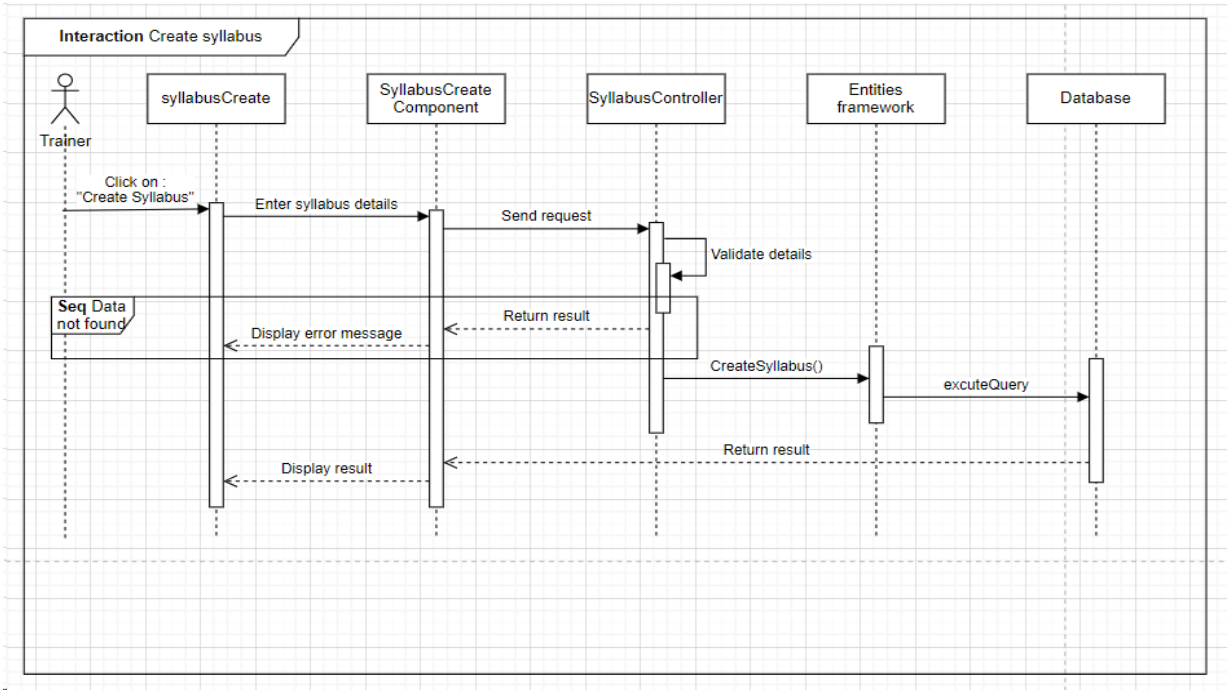


Figure 139 - <Super Admin, Class Admin, Trainer> Create Syllabus

4.3.2.5. <Super Admin, Class Admin, Trainer> Add unit training materials

Summary: This diagram show how the Super Admin, Class Admin, Trainer add unit training materials

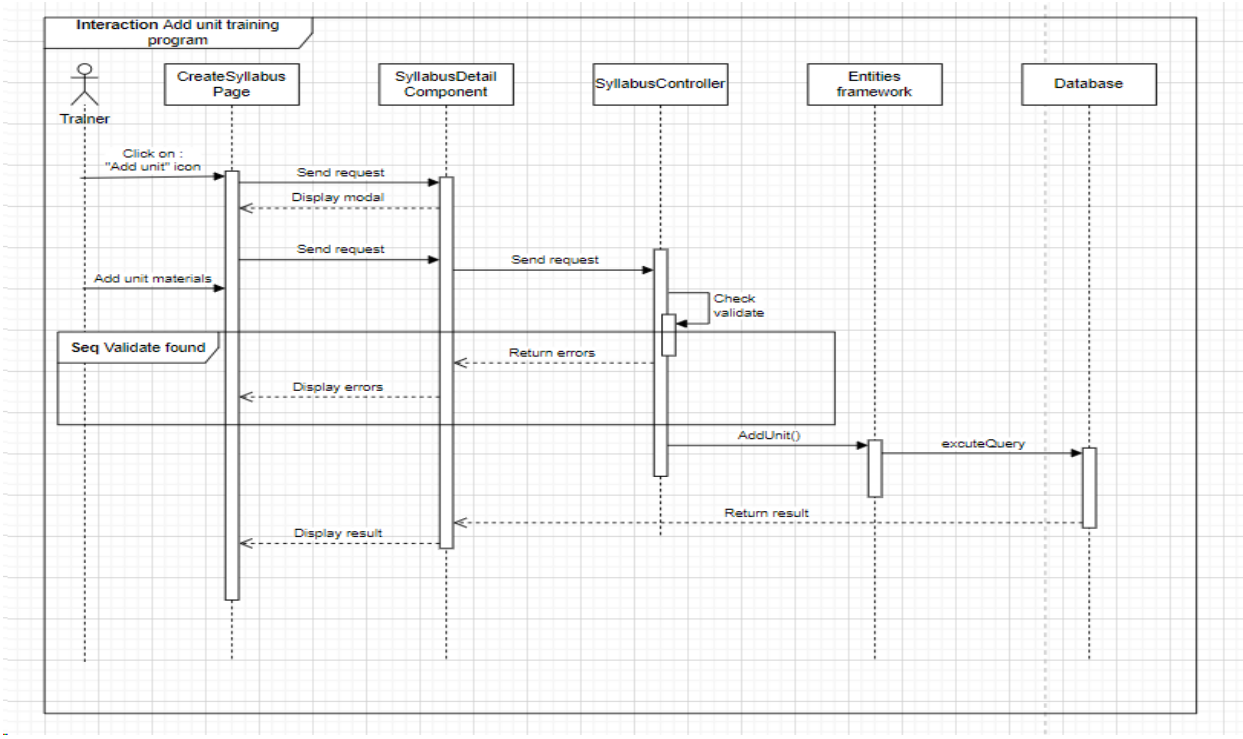


Figure 140 - <Super Admin, Class Admin, Trainer> Add unit training materials

4.3.2.6. <Super Admin, Class Admin, Trainer> Duplicate Syllabus

Summary: This diagram show how the Super Admin, Class Admin, Trainer duplicate Syllabus

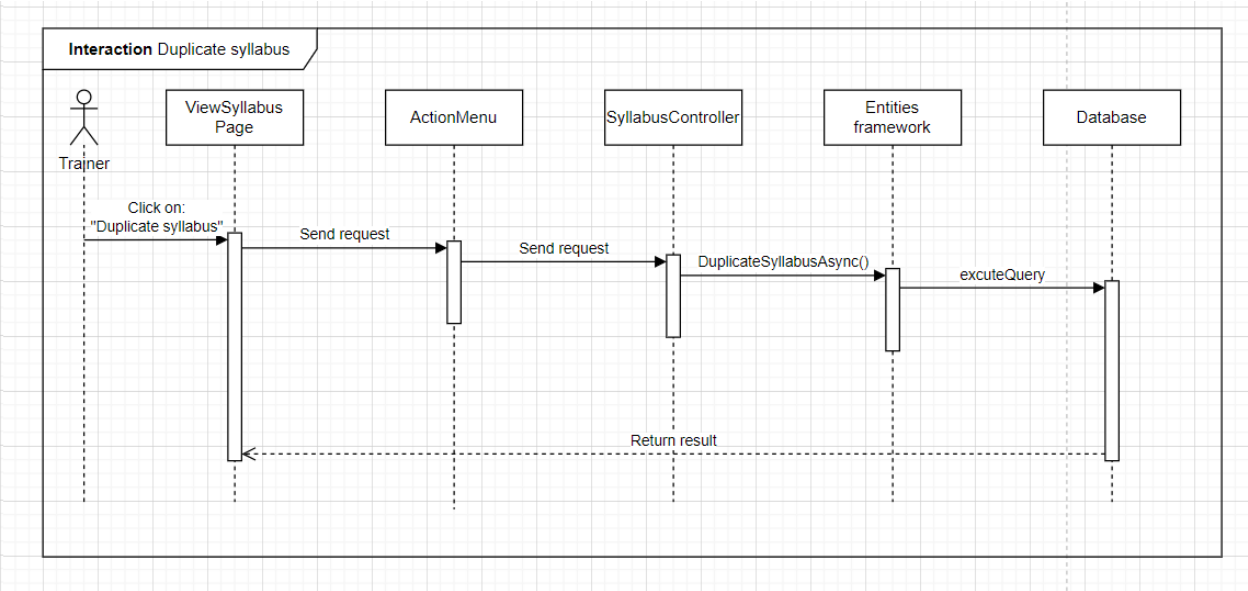


Figure 141 - <Super Admin, Class Admin, Trainer> Duplicate Syllabus

4.3.2.7. <Super Admin, Class Admin, Trainer> De-activate Syllabus

Summary: This diagram show how the Super Admin, Class Admin, Trainer de-activate Syllabus

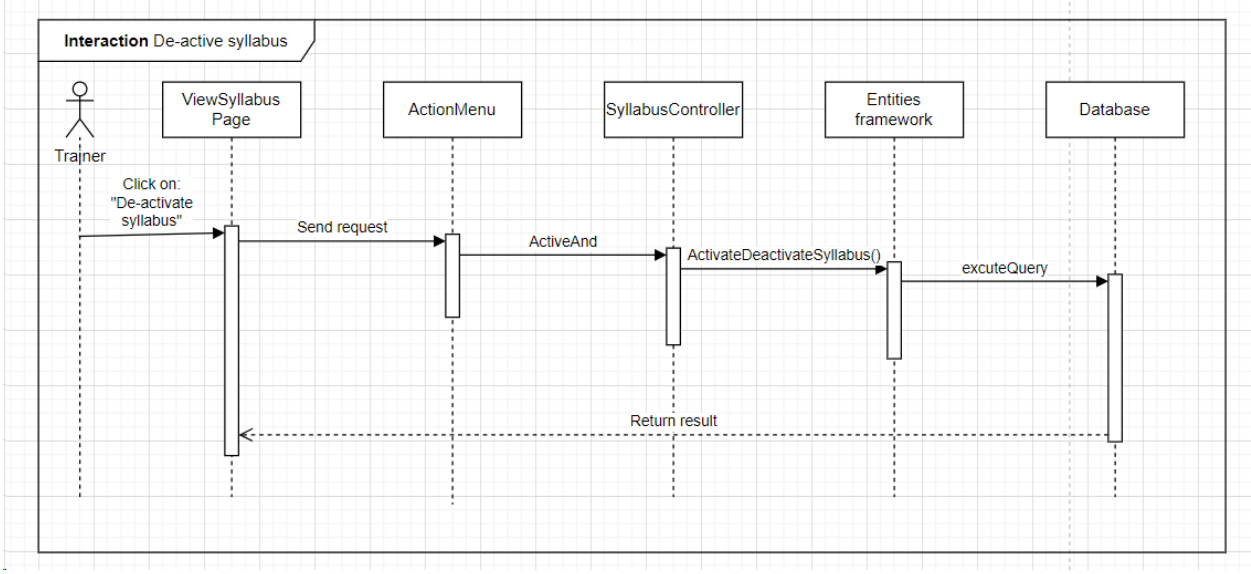


Figure 142 - <Super Admin, Class Admin, Trainer> De-activate Syllabus

4.3.2.8. <Super Admin, Class Admin, Trainer> Delete Syllabus

Summary: This diagram show how the Super Admin, Class Admin, Trainer delete Syllabus

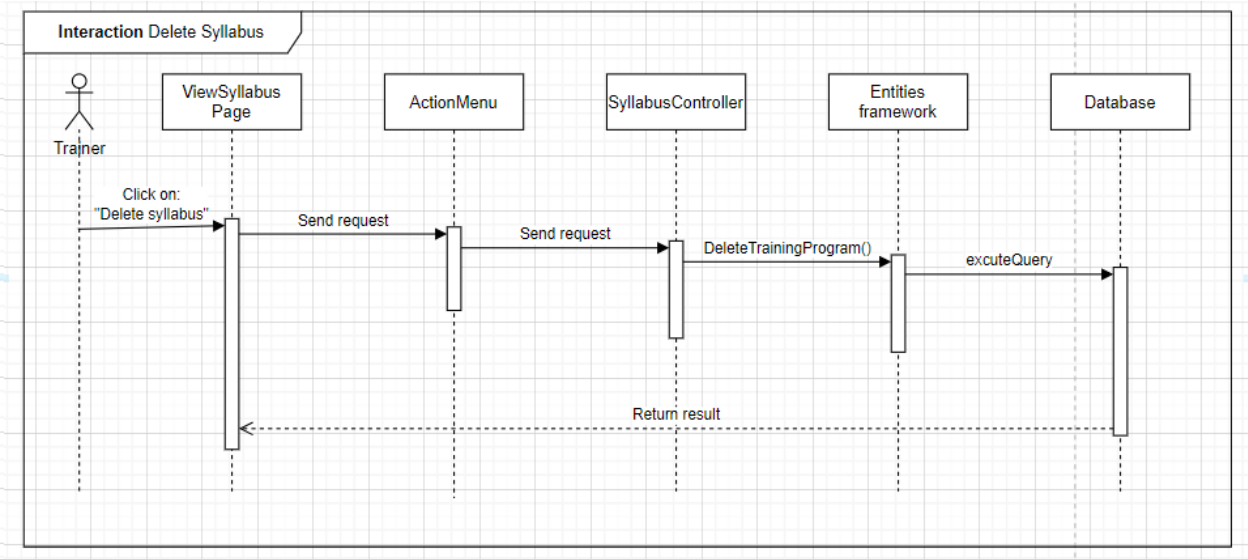


Figure 143 - <Super Admin, Class Admin, Trainer> Delete Syllabus

4.3.2.9. <Super Admin, Class Admin, Trainer> Edit Syllabus

Summary: This diagram show how the Super Admin, Class Admin, Trainer edit Syllabus

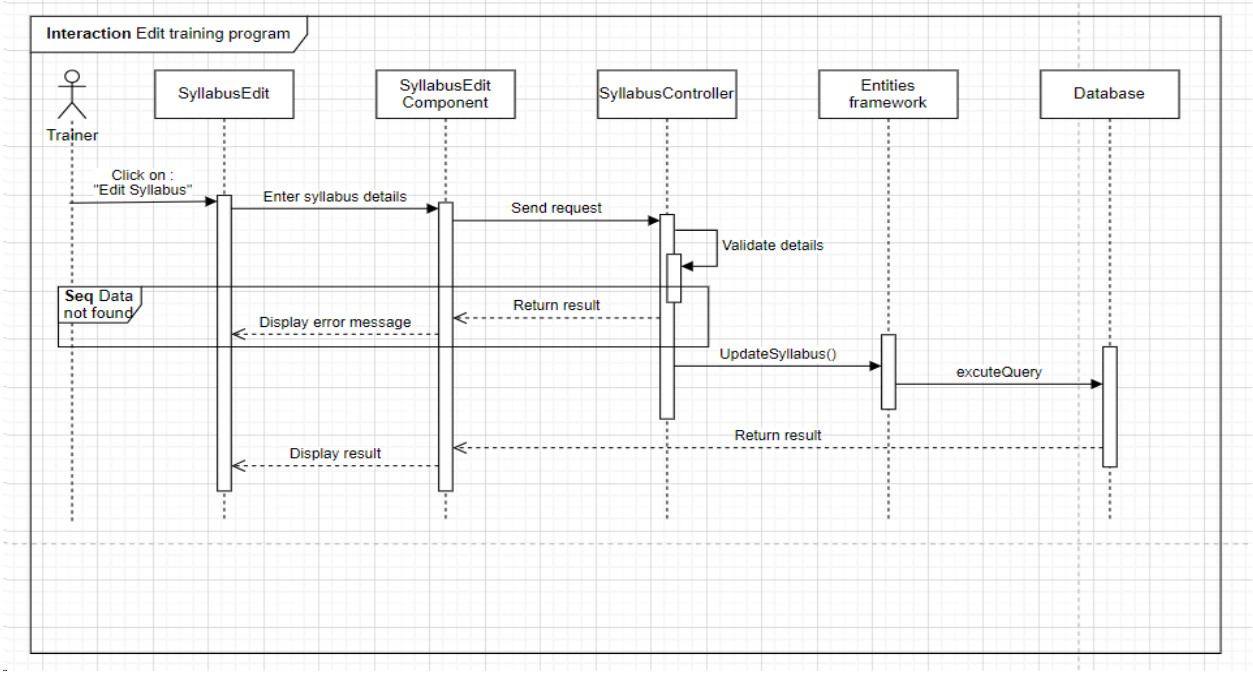


Figure 144 - <Super Admin, Class Admin, Trainer> Edit Syllabus

4.3.2.10. <Super admin, Class admin, Trainer> Export syllabus list

Summary: This diagram show how the Super admin, Class admin, Trainer export syllabus list

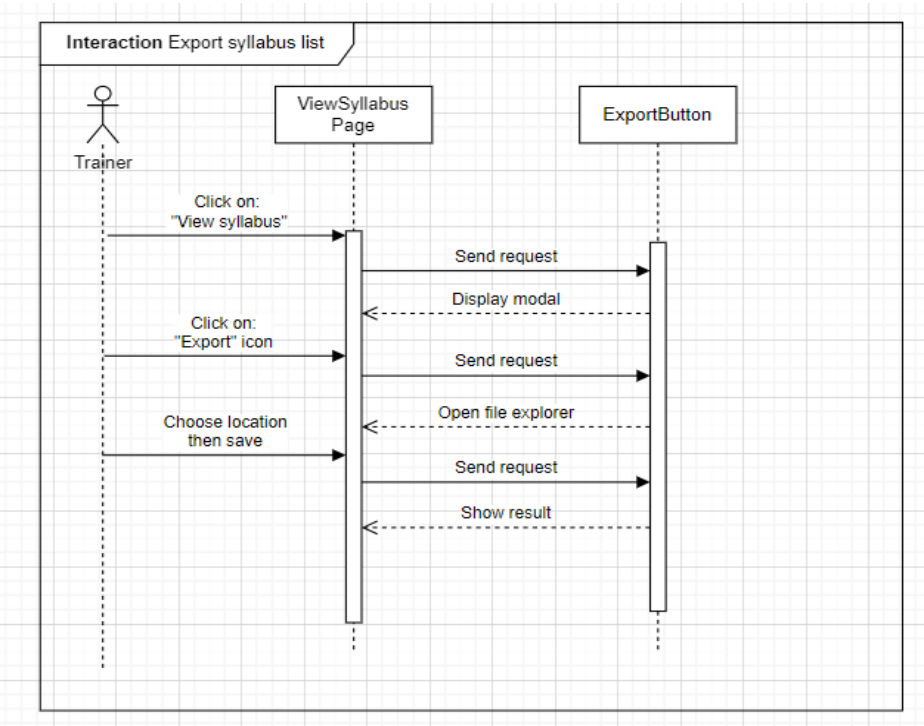


Figure 145 - <Super admin, Class admin, Trainer> Export syllabus list

4.3.3. Training Program Management Diagram

4.3.3.1. <Super Admin, Class Admin, Trainer> View list Training Program

Summary: This diagram show how the Super Admin, Class Admin, Trainer view list training program

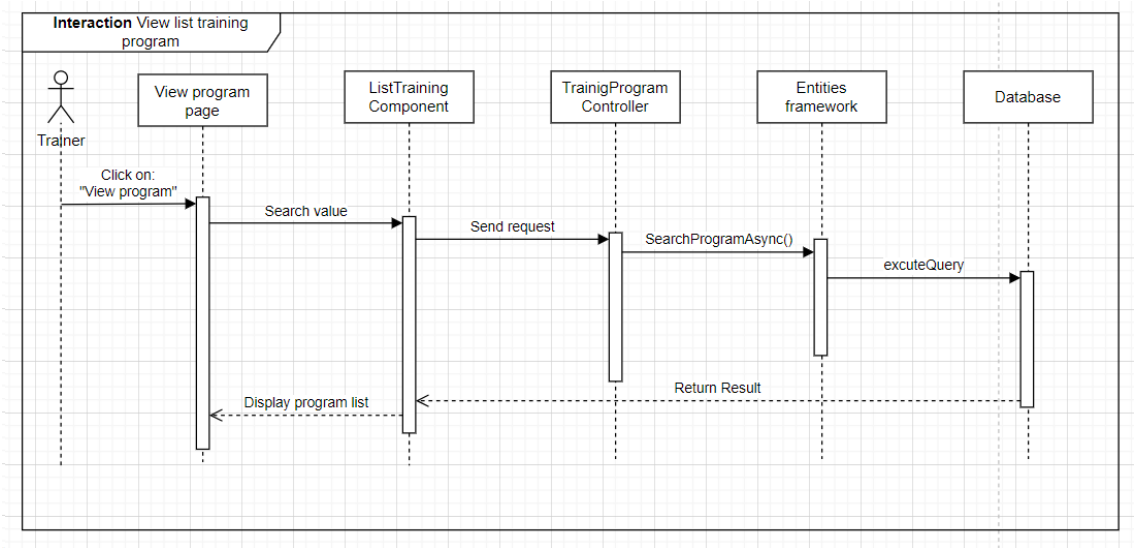


Figure 146 - <Super Admin, Class Admin, Trainer> View list Training Program

4.3.3.2. <Super Admin, Class Admin, Trainer> Filter training program

Summary: This diagram show how the Super Admin, Class Admin, Trainer filter training program

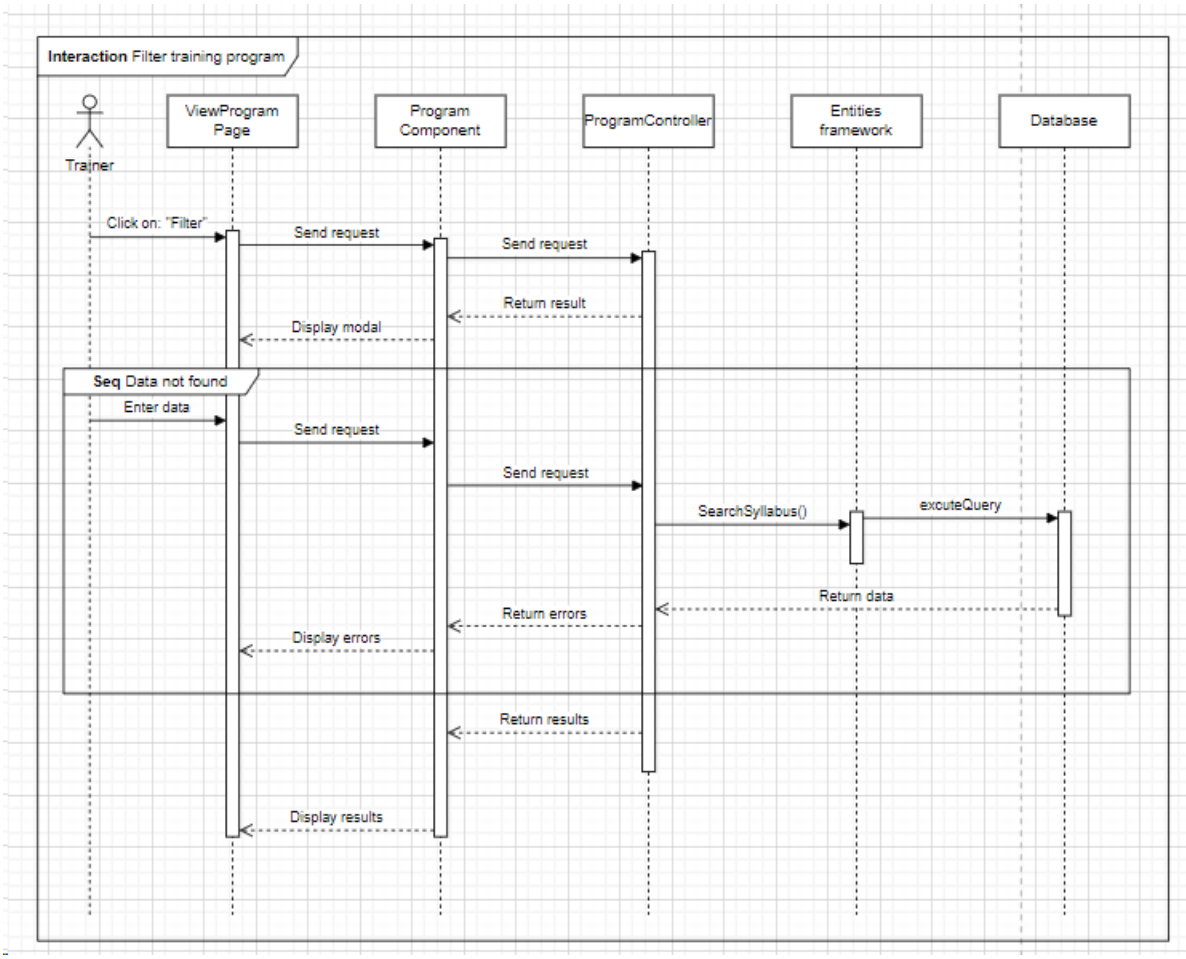


Figure 147 - <Super Admin, Class Admin, Trainer> Filter training program

4.3.3.3. <Super Admin, Class Admin> Edit training program

Summary: This diagram show how the Super Admin, Class Admin edit training program

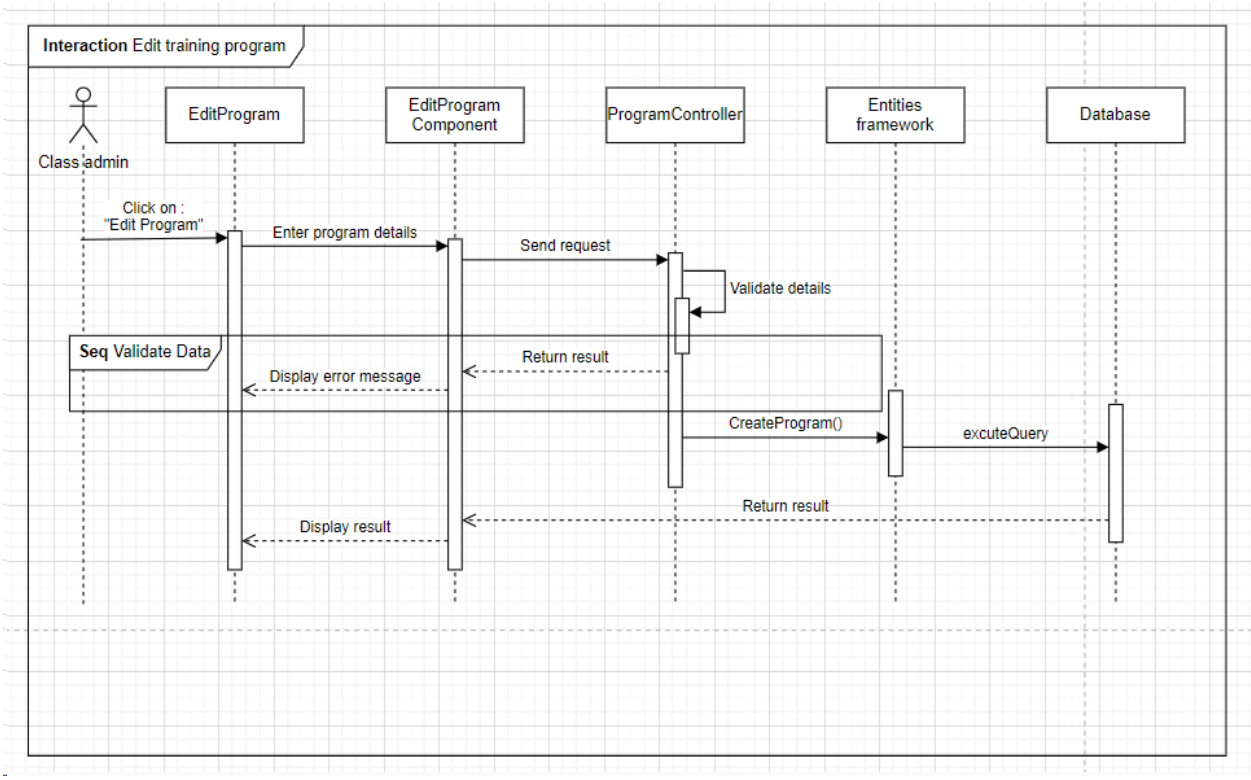


Figure 148 - <Super Admin, Class Admin> Edit training program

4.3.3.4. <Super Admin, Class Admin> Duplicate training program

Summary: This diagram show how the Super Admin, Class Admin duplicate training program

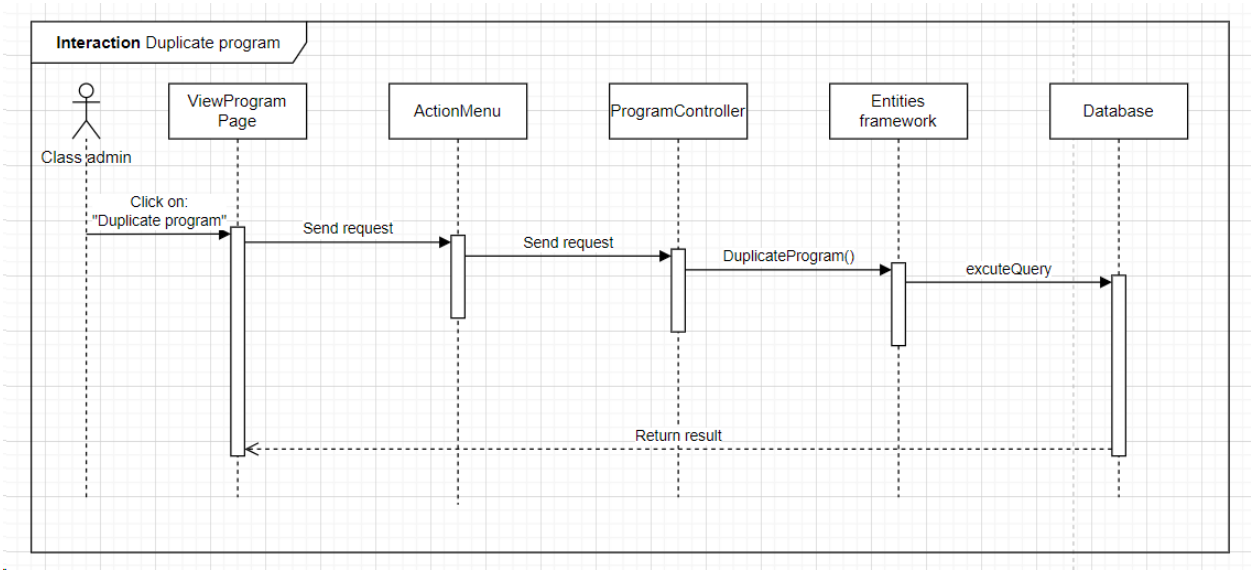


Figure 149 - <Super Admin, Class Admin> Duplicate training program

4.3.3.5. <Super Admin, Class Admin> Delete training program

Summary: This diagram show how the Super Admin, Class Admin delete training program

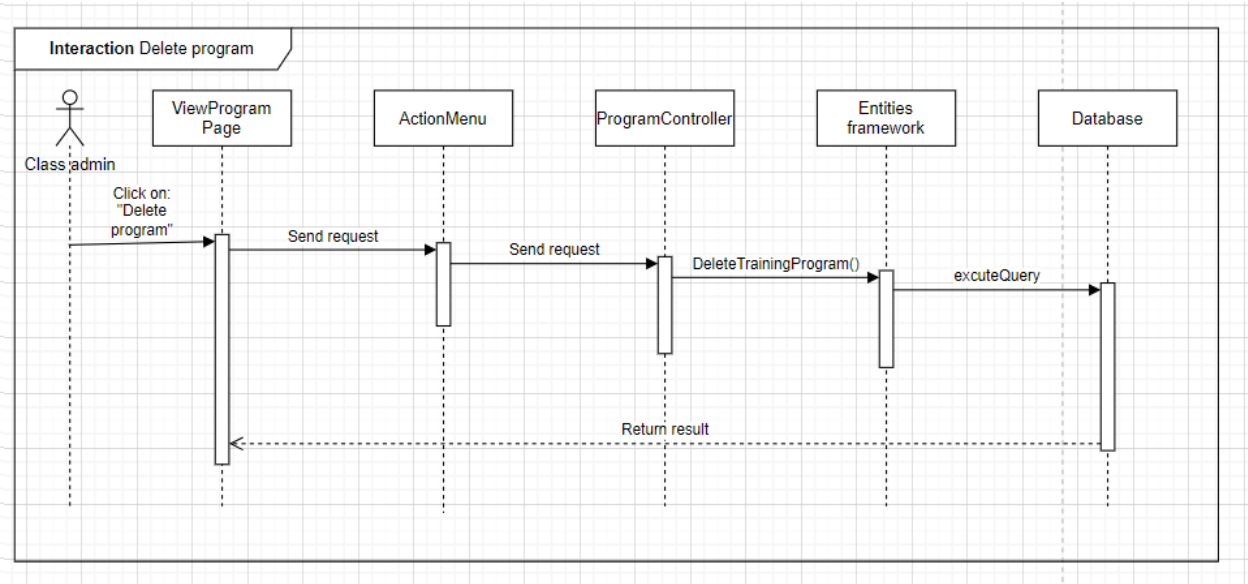


Figure 150 - <Super Admin, Class Admin> Delete training program

4.3.3.6. <Super Admin, Class Admin> De-activate training program

Summary: This diagram show how the Super Admin, Class Admin de-activate training program

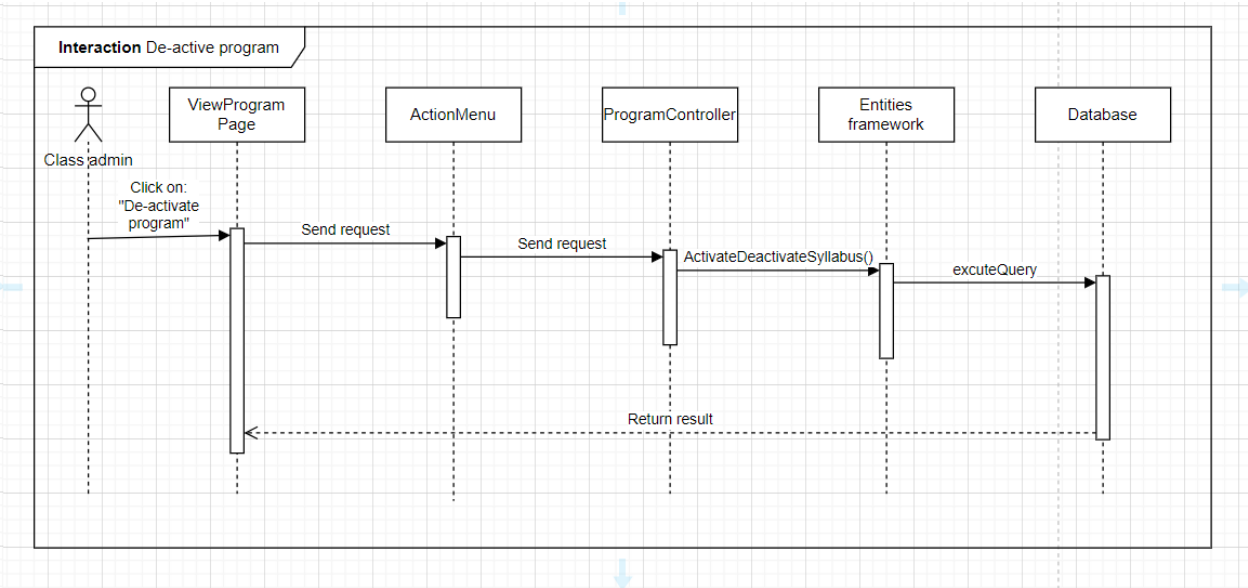


Figure 151 - <Super Admin, Class Admin> De-activate training program

4.3.3.7. <Super Admin, Class Admin>Import csv

Summary: This diagram show how the Super Admin, Class Admin import csv

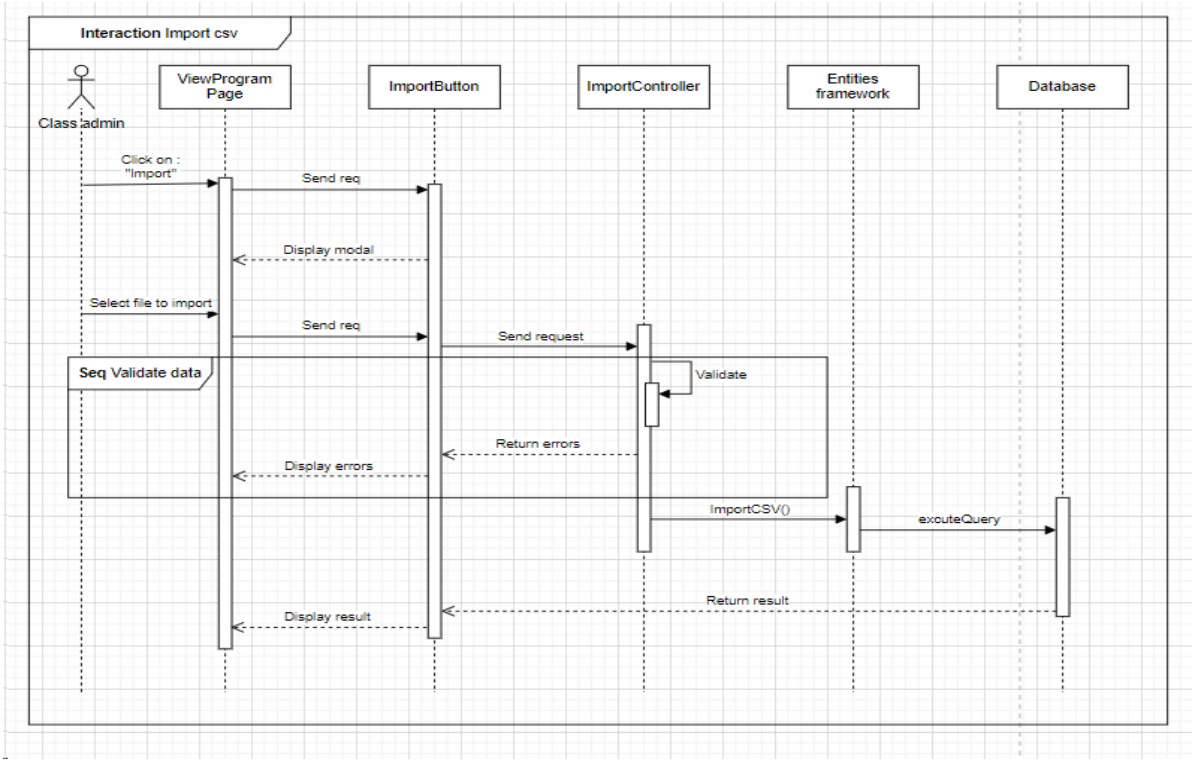


Figure 152 - <Super Admin, Class Admin> Import csv

4.3.3.8. <Super admin, Class admin, Trainer> Export training program

Summary: This diagram show how the Super admin, Class admin, Trainer export training program

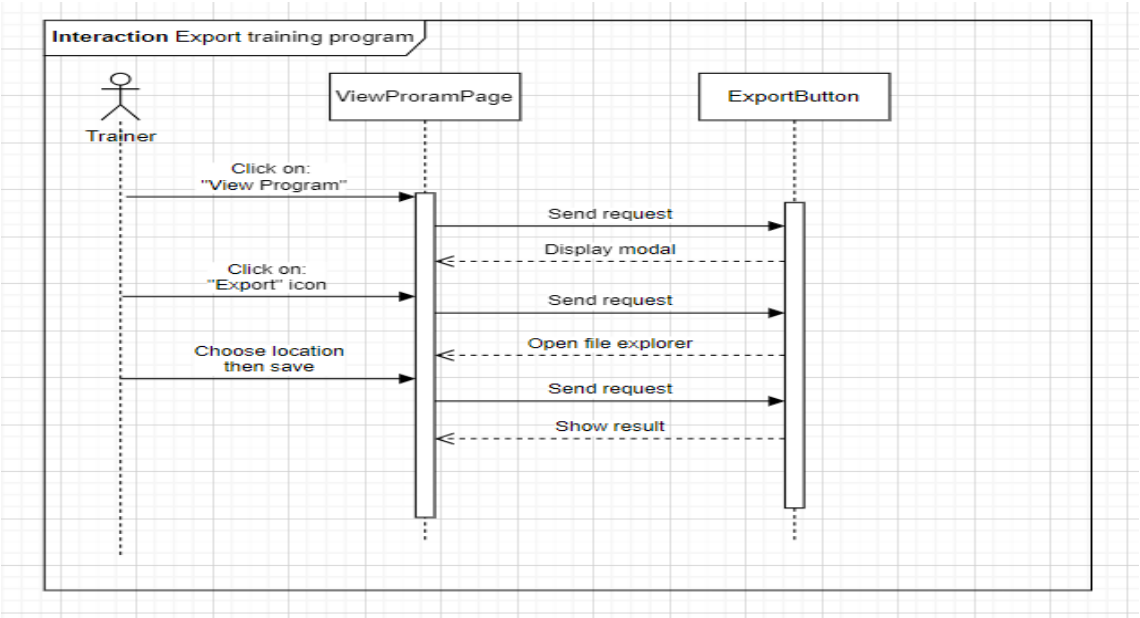


Figure 153 - <Super admin, Class admin, Trainer> Export training program

4.3.3.9. <Super Admin, Class Admin> View syllabus training materials

Summary: This diagram show how the Super Admin, Class Admin view syllabus training materials

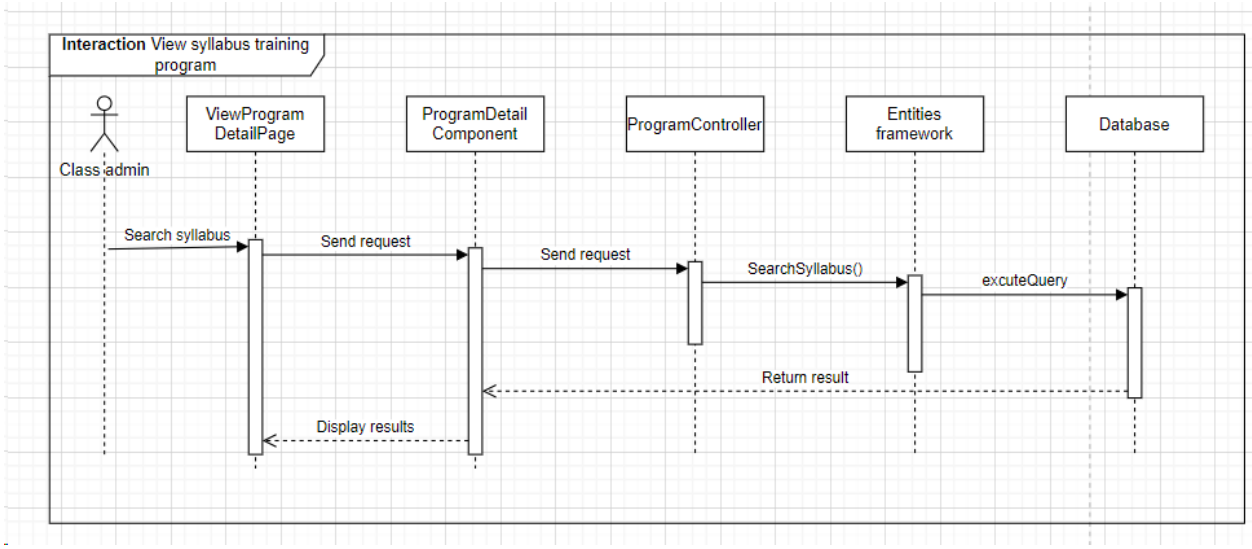


Figure 154 - <Super Admin, Class Admin> View syllabus training program

4.3.3.10. <Super Admin, Class Admin> Create training program

Summary: This diagram show how the Super Admin, Class Admin create training program

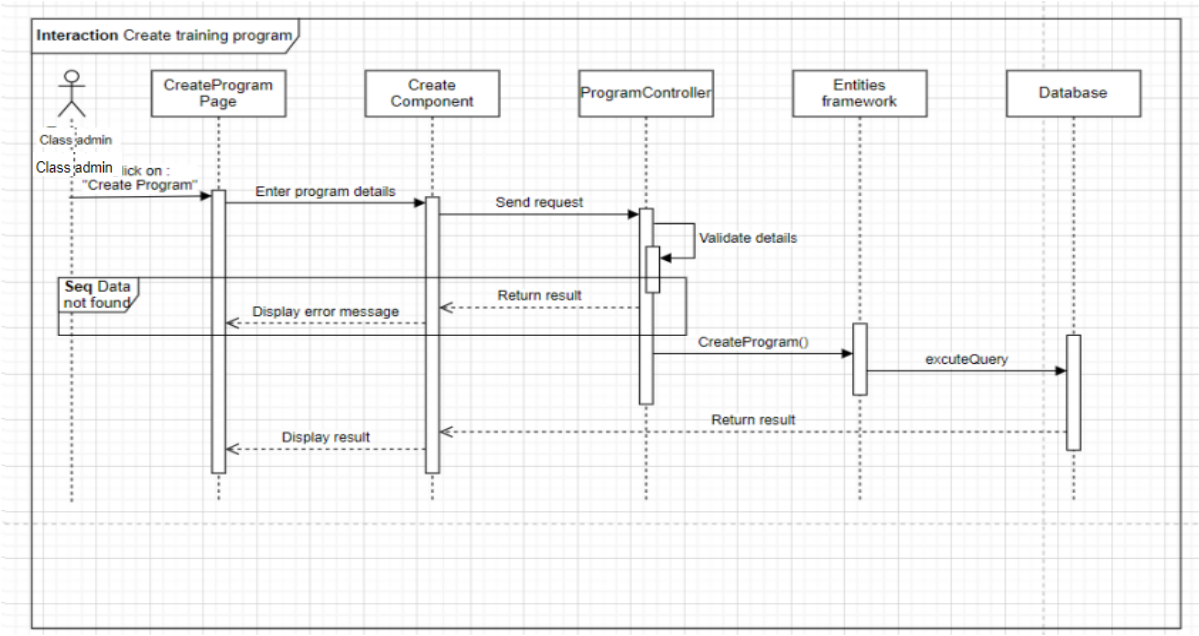


Figure 155 - <Super admin, Class admin> Create training program

4.3.3.11. <Super Admin, Class Admin> Upload new training program materials

Summary: This diagram show how the Super Admin, Class Admin upload new training program materials

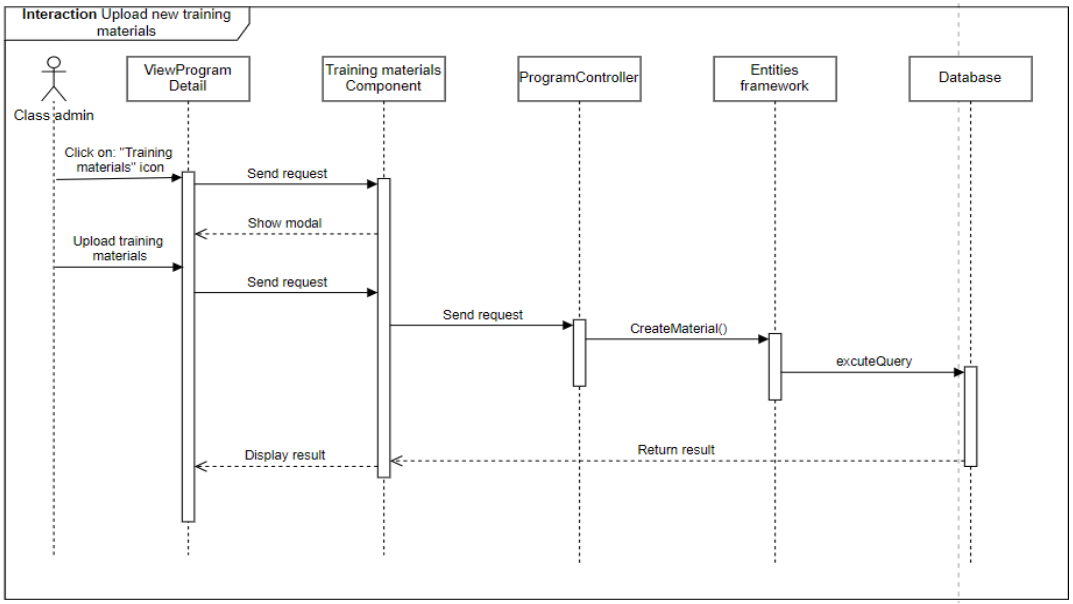


Figure 156 - <Super admin, Class admin> Upload new training program materials

4.3.3.12. <Super Admin, Class Admin> Delete training program materials

Summary: This diagram show how the Super Admin, Class Admin delete training program materials

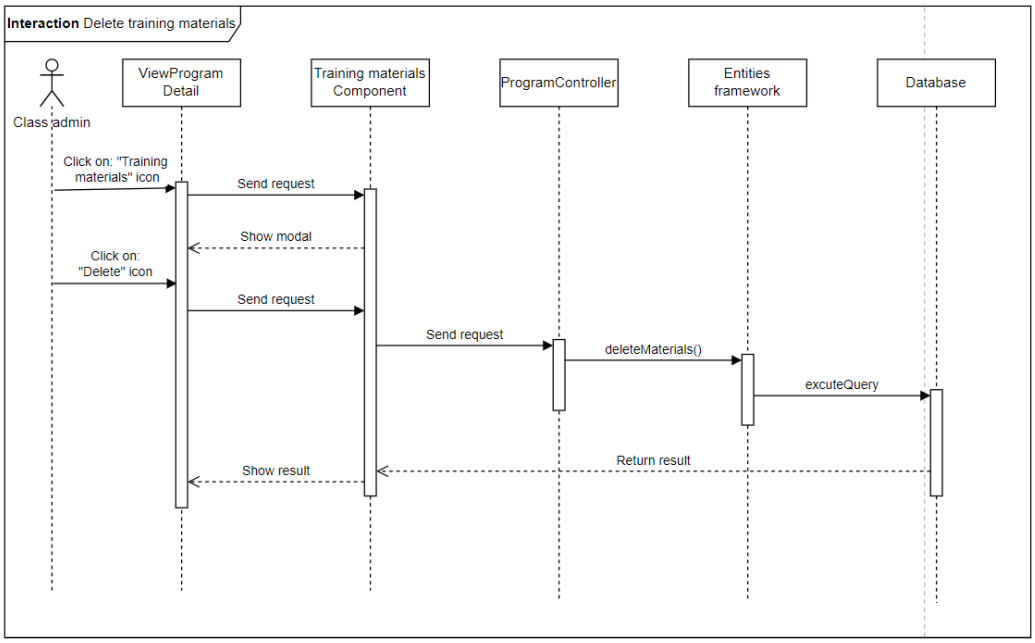


Figure 157 - <Super admin, Class admin> Delete training program materials

4.3.3.13. <Super Admin, Class Admin> Edit training program materials

Summary: This diagram show how the Super Admin, Class Admin edit training program materials

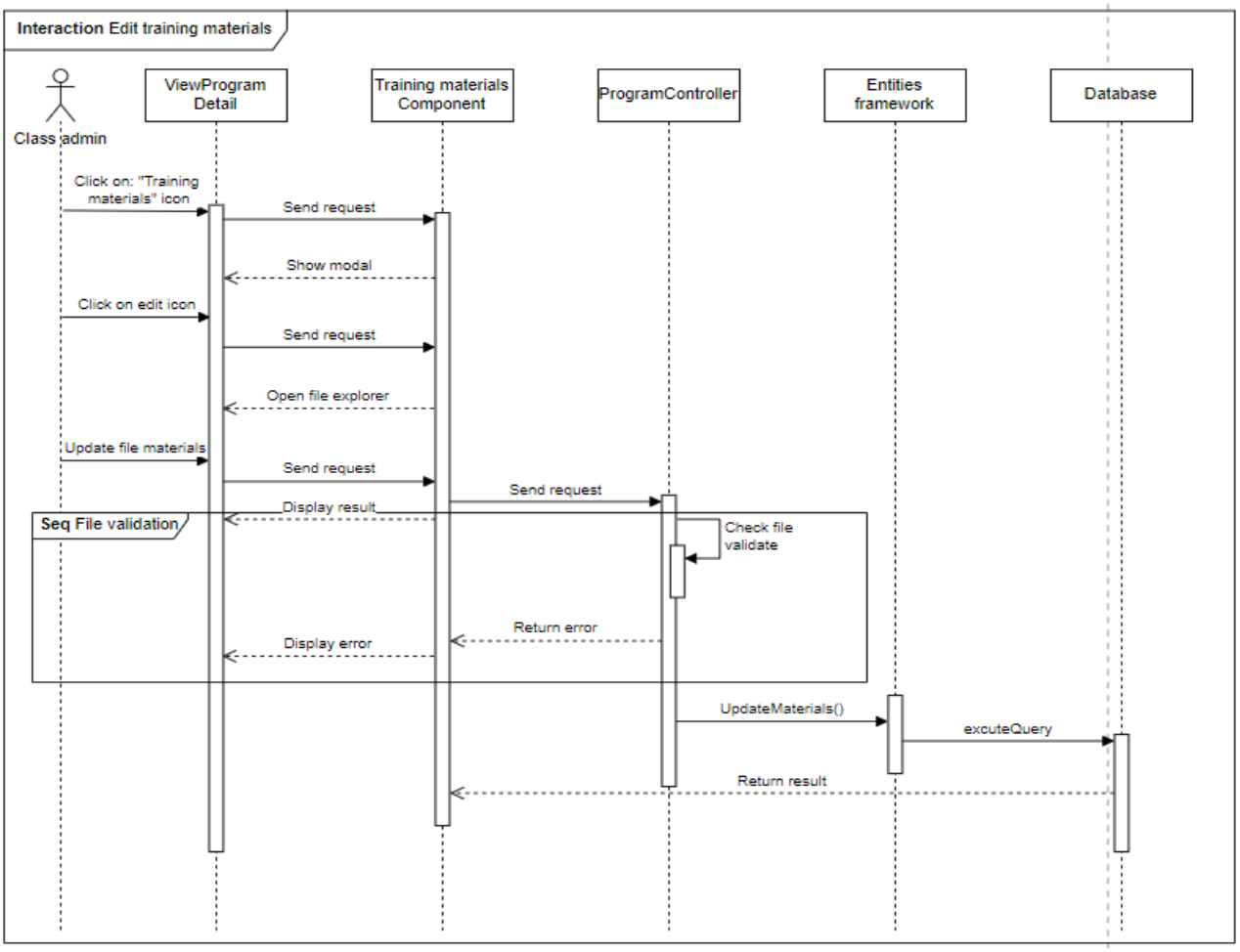


Figure 158 - <Super admin, Class admin> Edit training program materials

4.3.4. Class Management Diagram

4.3.4.1. <Super Admin, Class Admin, Trainer> View Training Calendar

Summary: This diagram show how the Super Admin, Class Admin, Trainer view training calendar

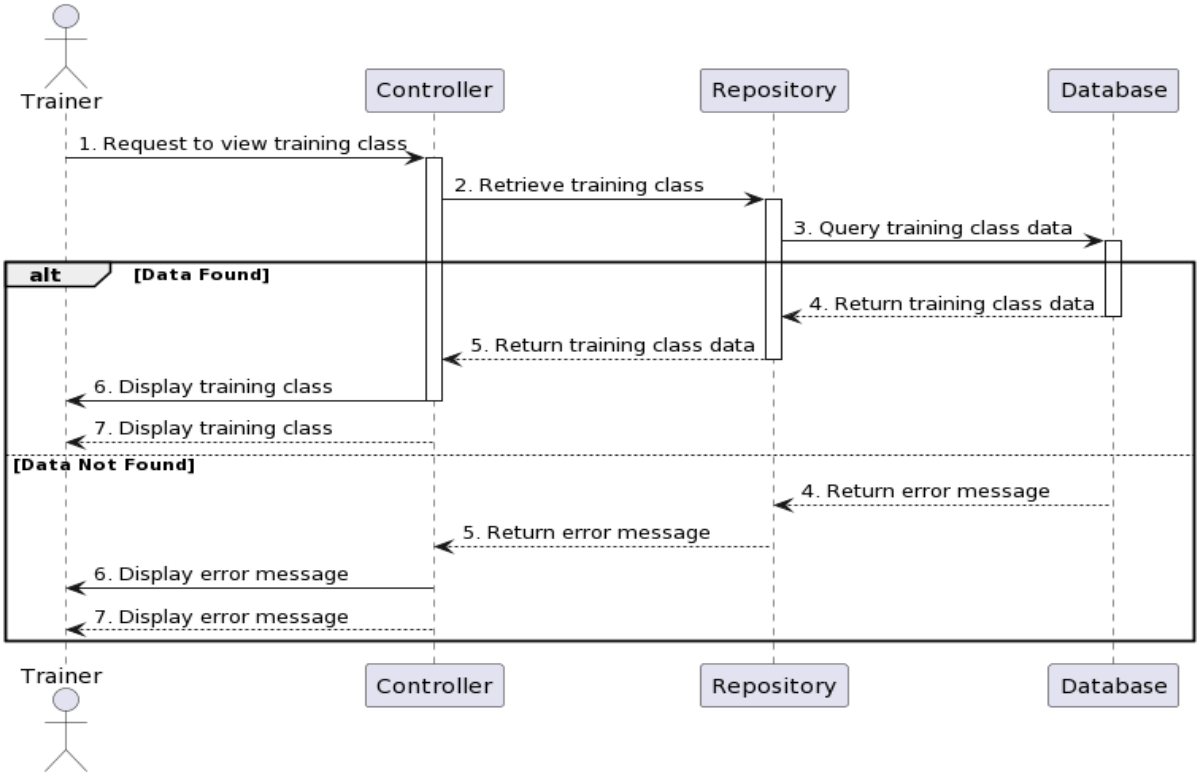


Figure 159 - <Super Admin, Class Admin, Trainer> View Training Calendar

4.3.4.2. <Super Admin, Class Admin, Trainer> View Class List

Summary: This diagram show how the Super Admin, Class Admin, Trainer view class list

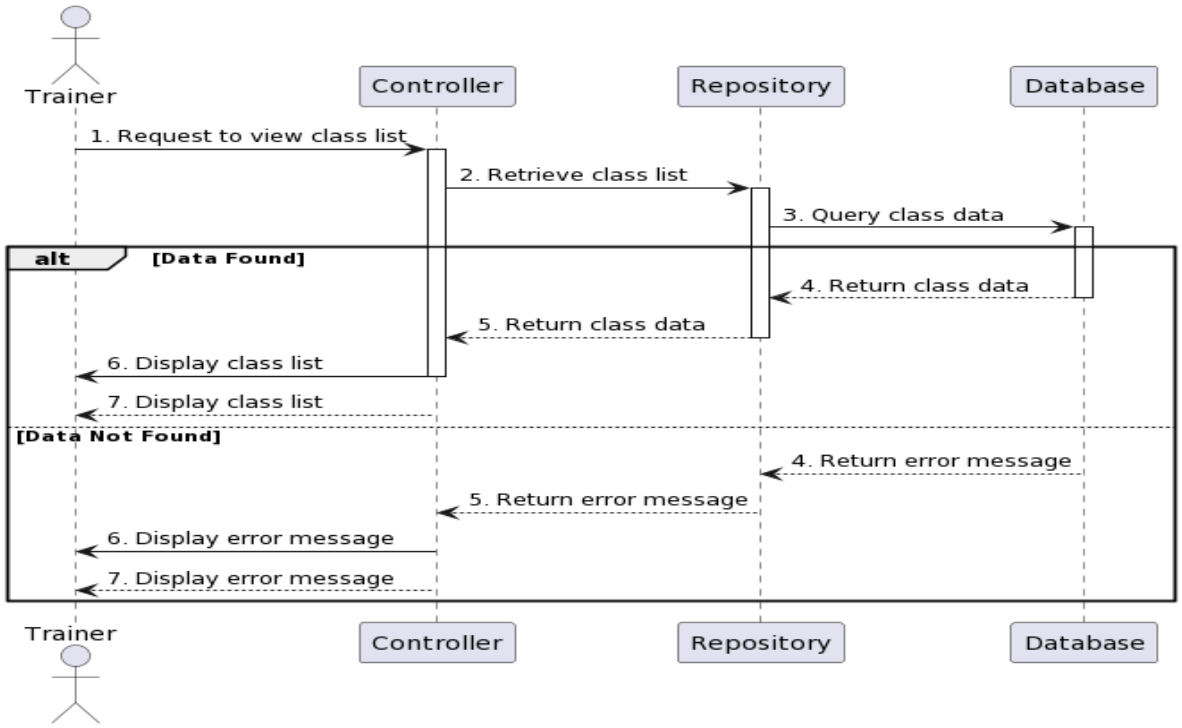


Figure 160 - <Super Admin, Class Admin, Trainer> View Class List

4.3.4.3. <Super Admin, Class Admin> Create Class

Summary: This diagram show how the Super Admin, Class Admin create class

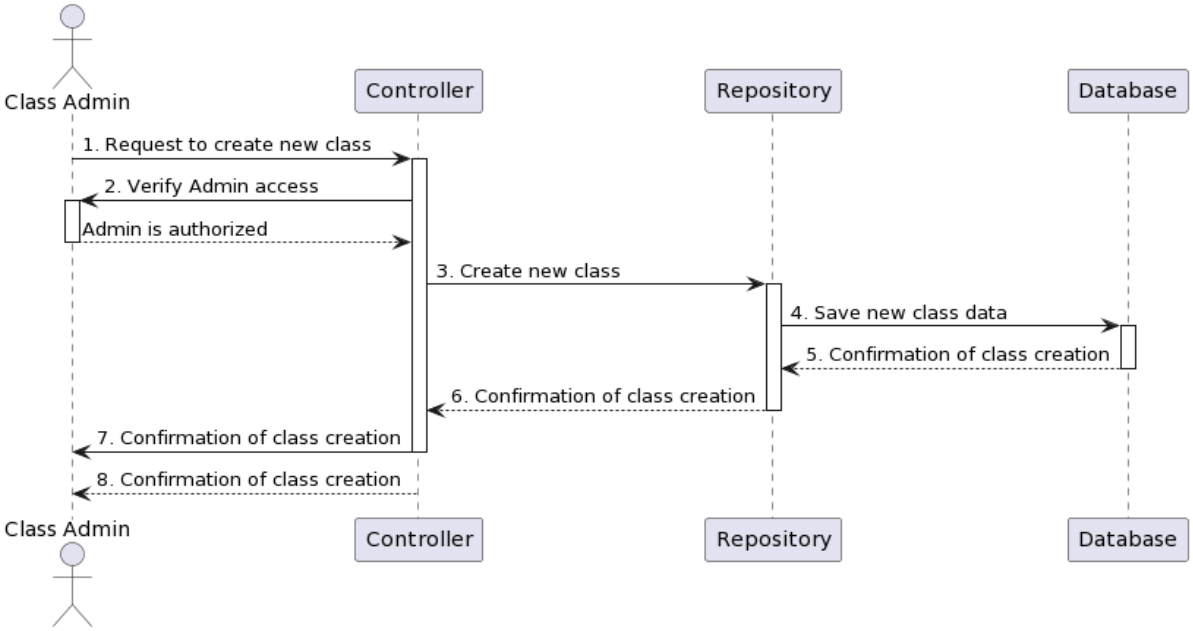


Figure 161 - <Super Admin, Class Admin> Create Class

4.3.4.4. <Super Admin, Class Admin, Trainer> View Detail Class

Summary: This diagram show how the Super Admin, Class Admin, Trainer view detail class

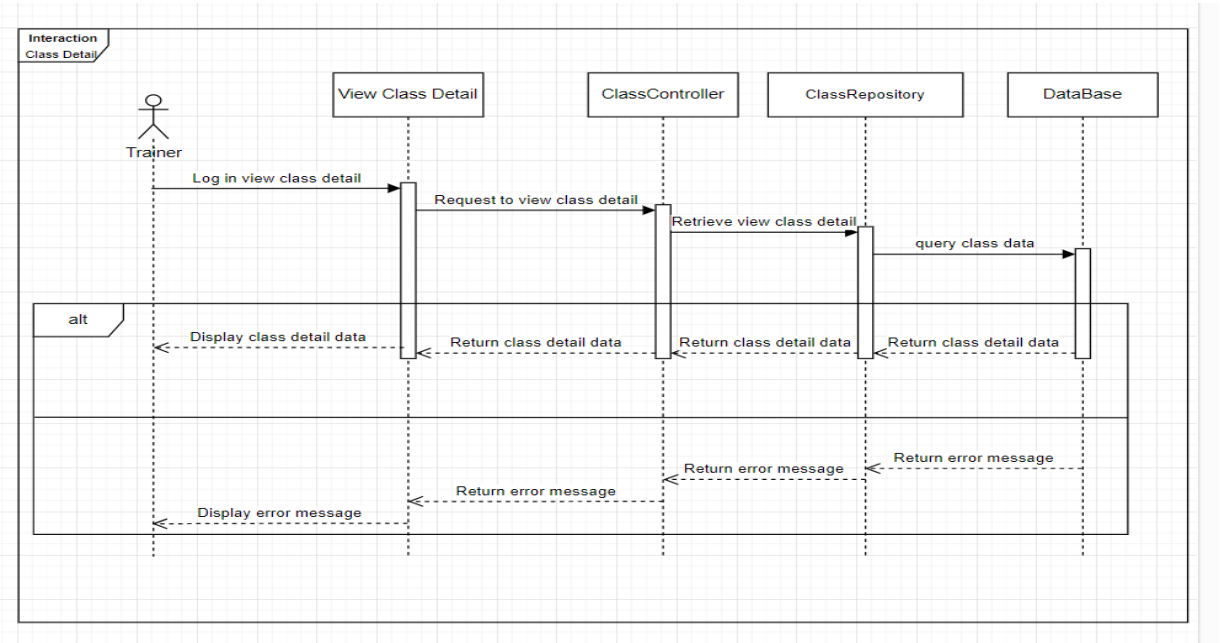


Figure 162 - <Super Admin, Class Admin, Trainer> View Detail List

4.3.4.5. <Super Admin, Class Admin, Trainer> Filter class list

Summary: This diagram show how the Super Admin, Class Admin, Trainer filter class list

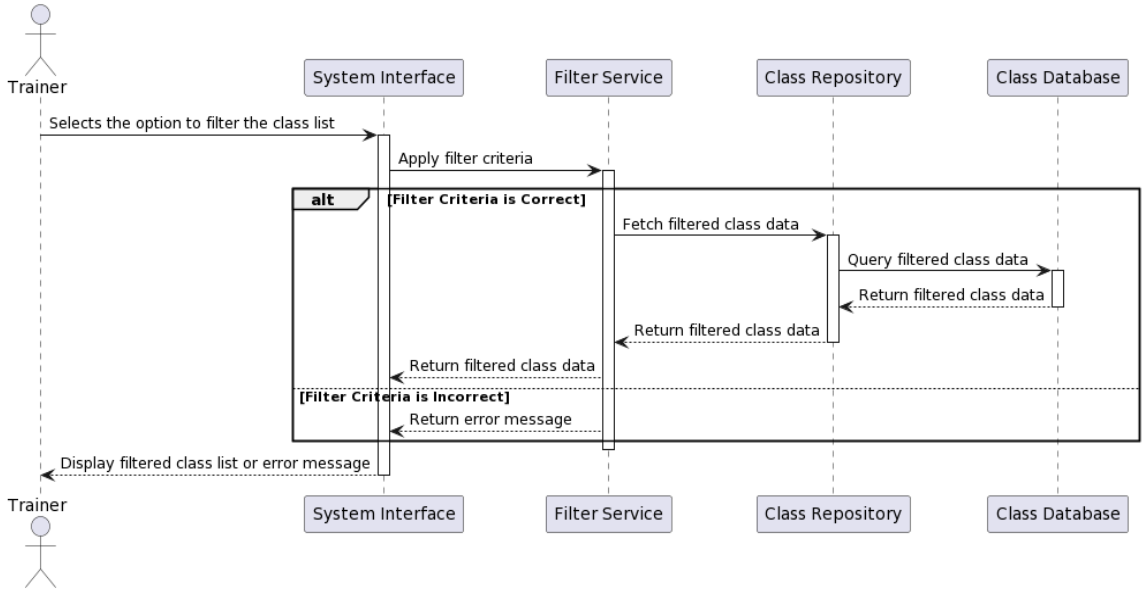


Figure 163 - <Super Admin, Class Admin, Trainer> Filter Class List

4.3.4.6. <Super Admin, Class Admin> Delete Syllabus

Summary: This diagram show how the Super Admin, Class Admin delete syllabus

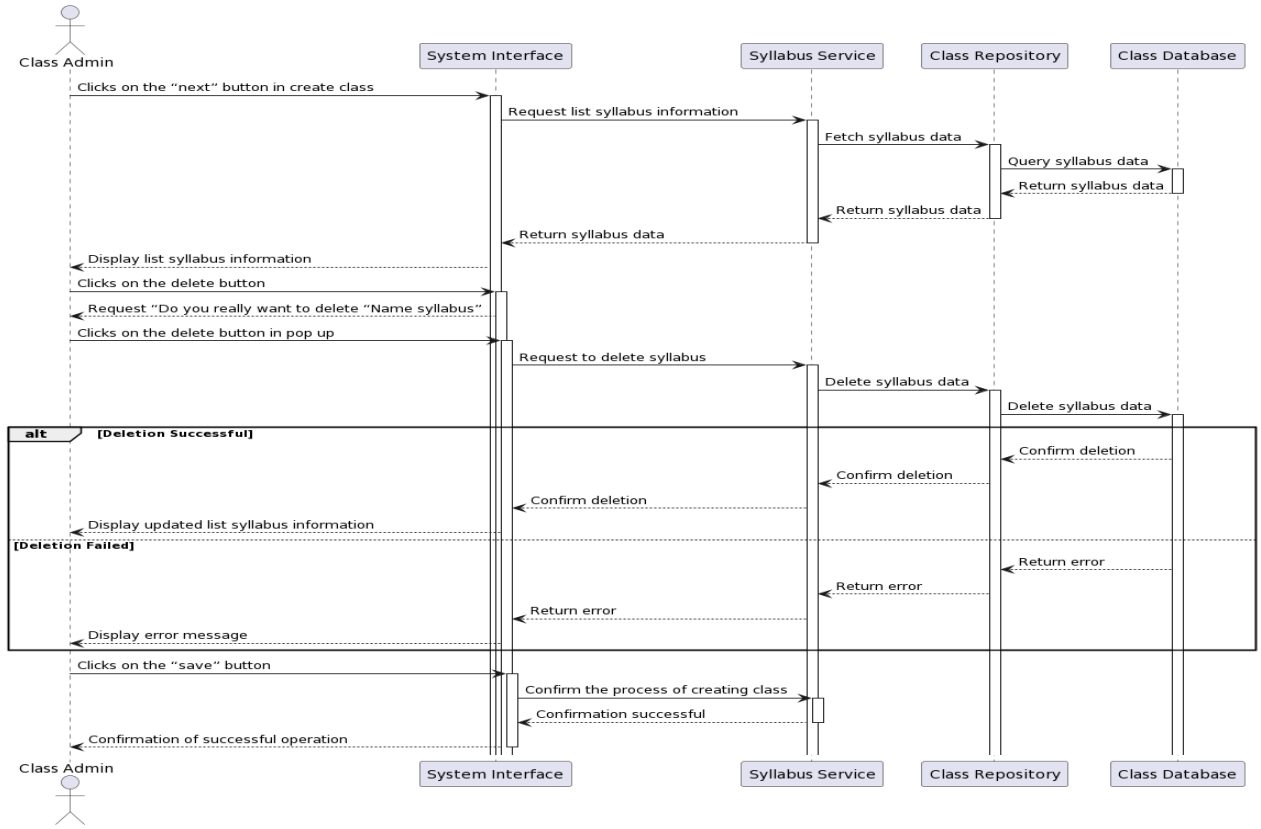


Figure 164 - <Super Admin, Class Admin> Delete Syllabus

4.3.4.7. <Super Admin, Class Admin, Trainer> Search Class on List

Summary: This diagram show how the Super Admin, Class Admin, Trainer search class on list

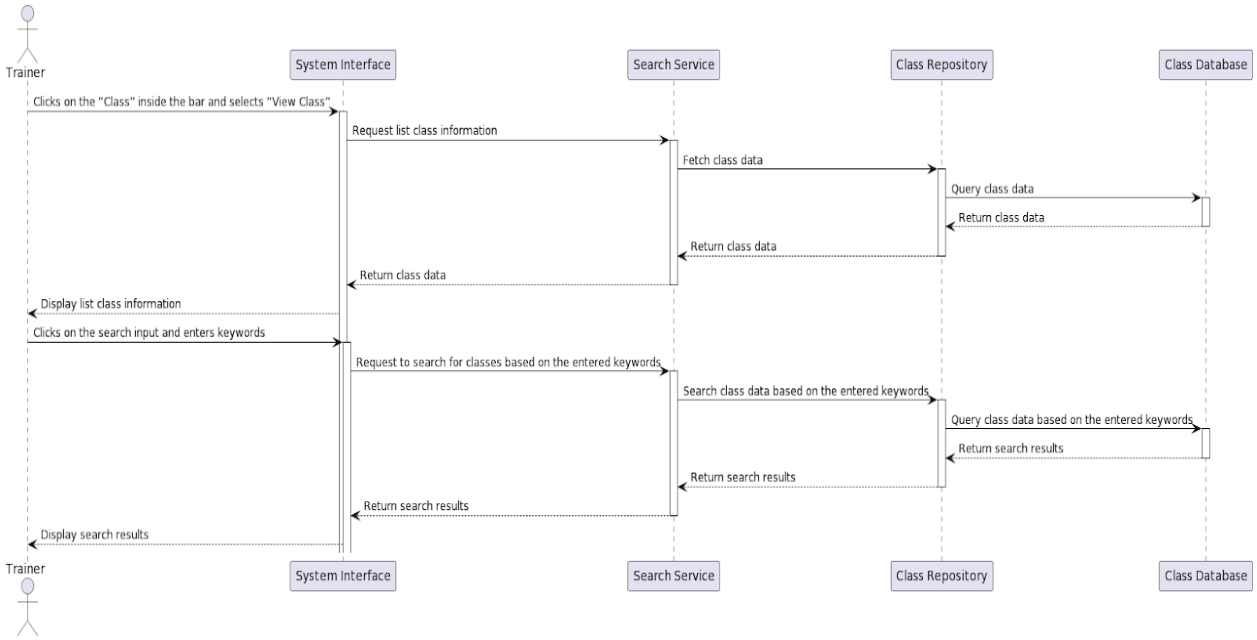


Figure 165 - <Super Admin, Class Admin, Trainer> Search Class on List

4.3.4.8. <Super Admin, Class Admin> Edit Class

Summary: This diagram show how the Super Admin, Class Admin edit class

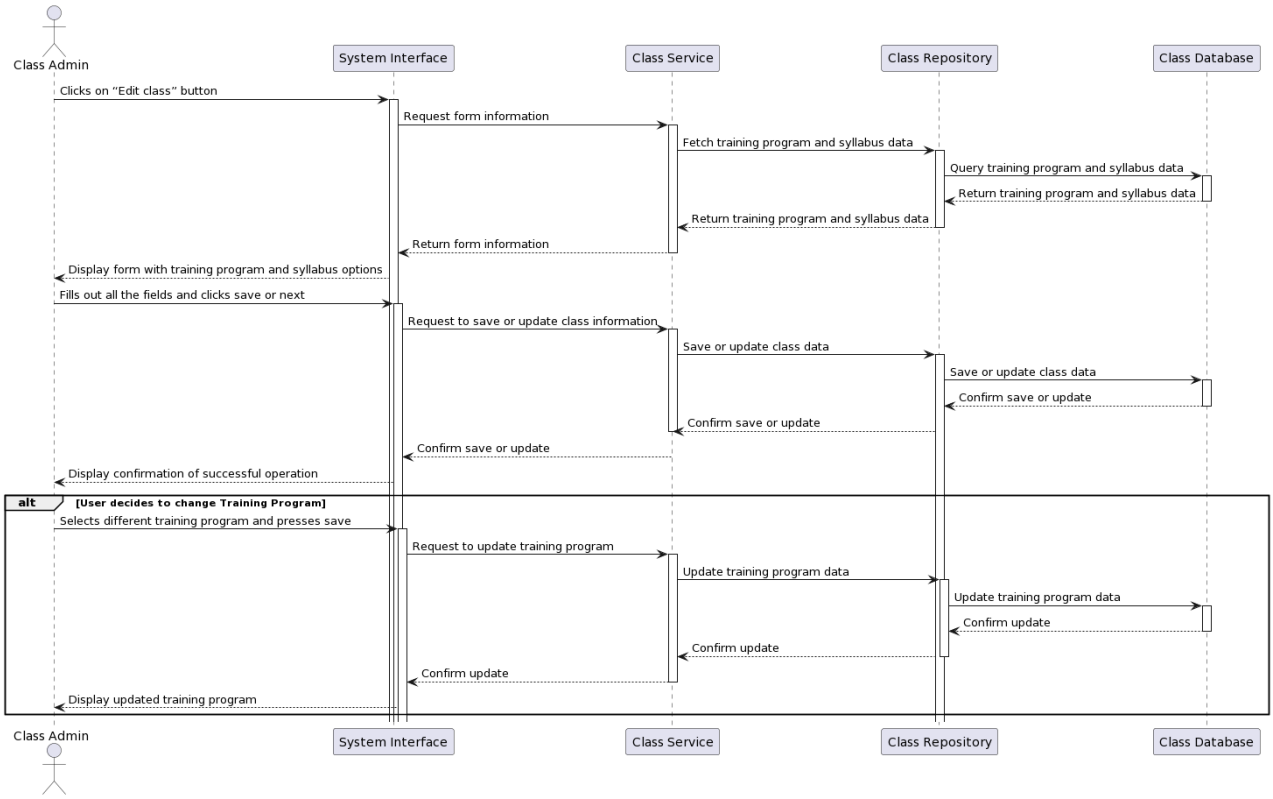


Figure 166 - <Super Admin, Class Admin> Edit Class

4.3.4.9. <Super Admin, Class Admin, Trainer> Search Calendar by Week

Summary: This diagram show how the Super Admin, Class Admin, Trainer search calendar by week

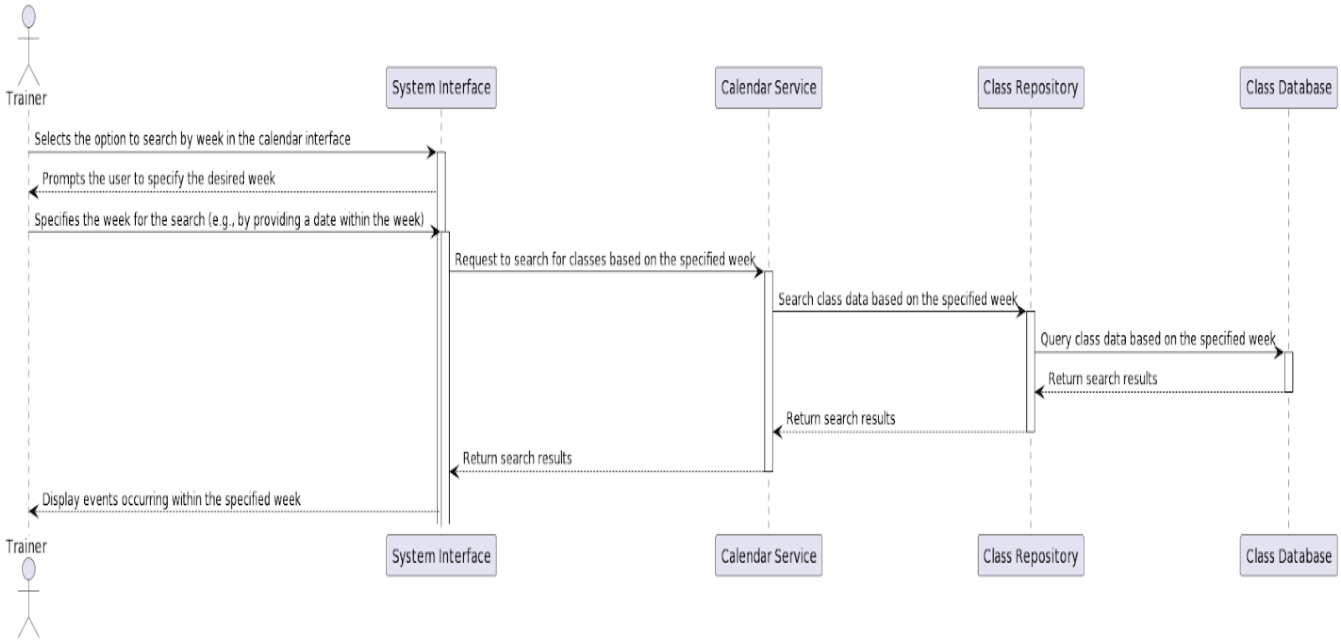


Figure 167 - <Super Admin, Class Admin, Trainer> Search Calendar by Week

4.3.4.10. <Super Admin, Class Admin, Trainer> Filter Calendar by Day

Summary: This diagram show how the Super Admin, Class Admin, Trainer Filter Calendar by Day

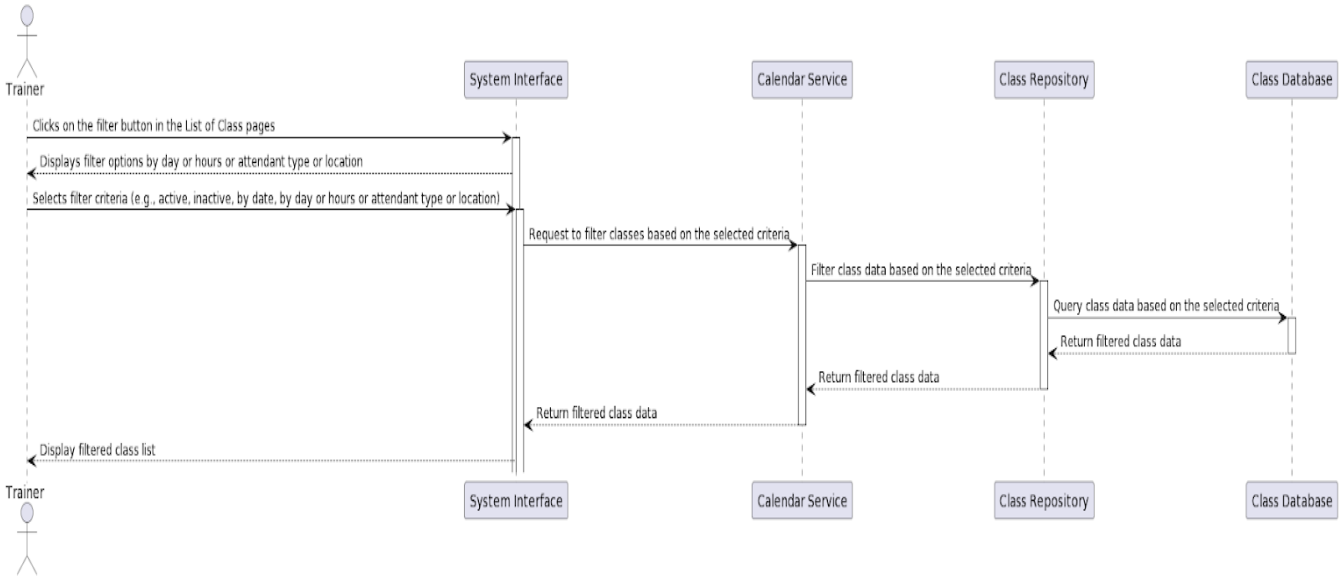


Figure 168 - <Super Admin, Class Admin, Trainer> Filter Calendar by Day

4.3.4.11. <Super Admin, Class Admin, Trainer> Sort Class

Summary: This diagram show how the Super Admin, Class Admin, Trainer sort class

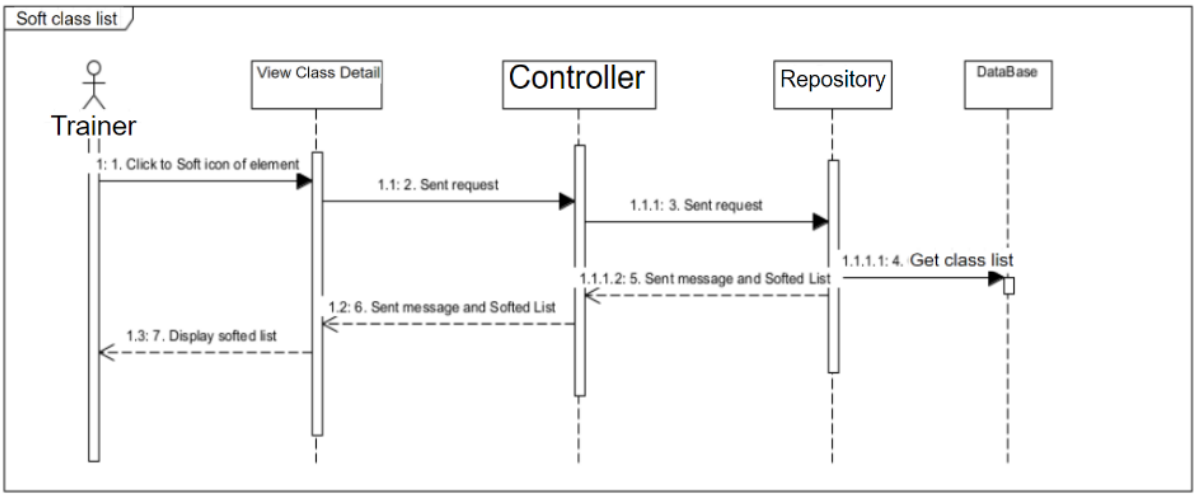


Figure 169 - <Super Admin, Class Admin, Trainer> Sort Class

4.3.4.12. <Super Admin, Class Admin> Duplicated Class

Summary: This diagram show how the Super Admin, Class Admin view duplicated class

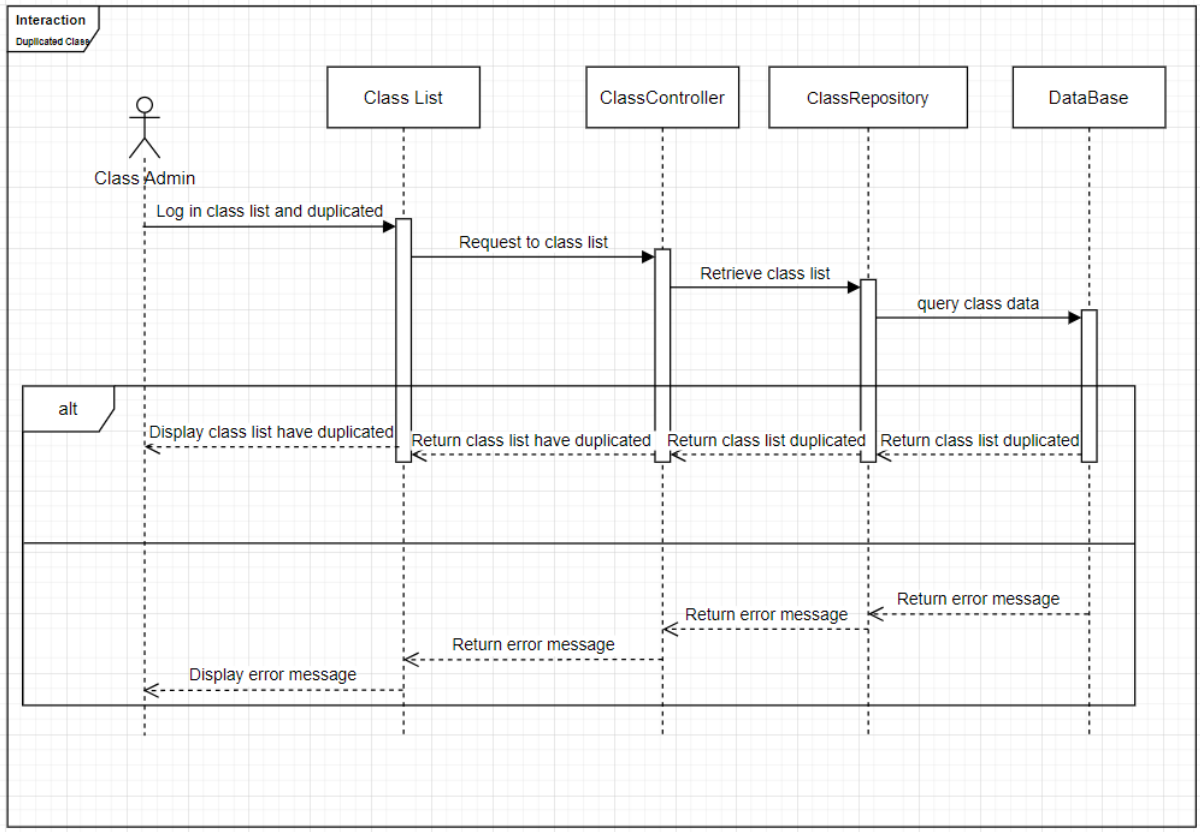


Figure 170 - <Super Admin, Class Admin> Duplicated Class

4.3.4.13. <Super Admin, Class Admin, Trainer> Elasticsearch

Summary: This diagram show how the Super Admin, Class Admin, Trainer Elastic search

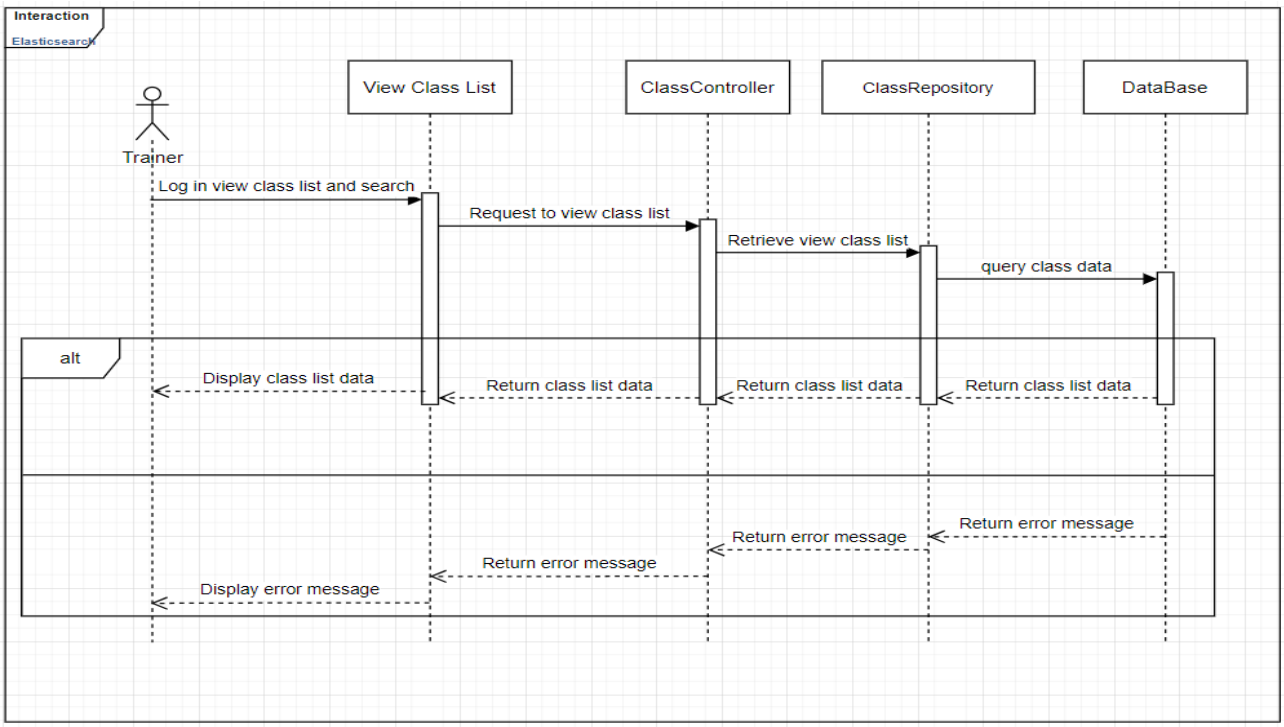


Figure 171 - <Super Admin, Class Admin, Trainer> Elasticsearch

4.3.4.14. <Super Admin, Class Admin, Trainer> Search Class

Summary: This diagram show how the Super Admin, Class Admin, Trainer search class

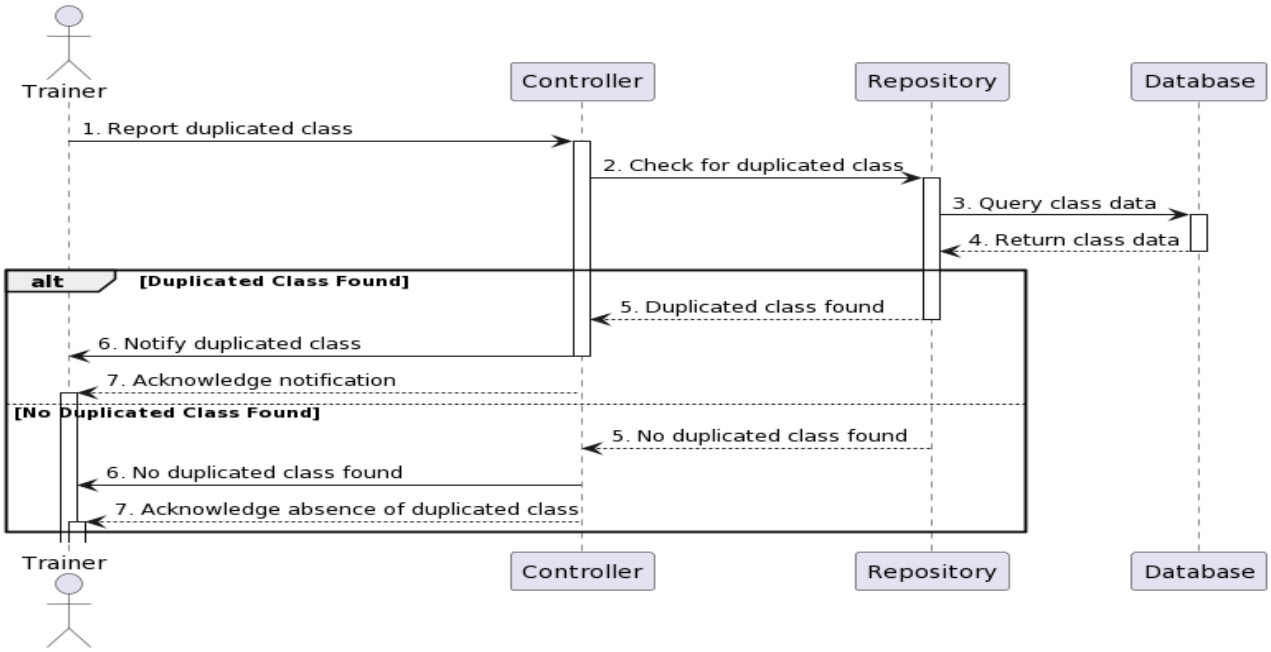


Figure 172 - <Super Admin, Class Admin, Trainer> Search Class

4.3.4.15. <Super Admin, Class Admin> Delete Class

Summary: This diagram show how the Super Admin, Class Admin delete class

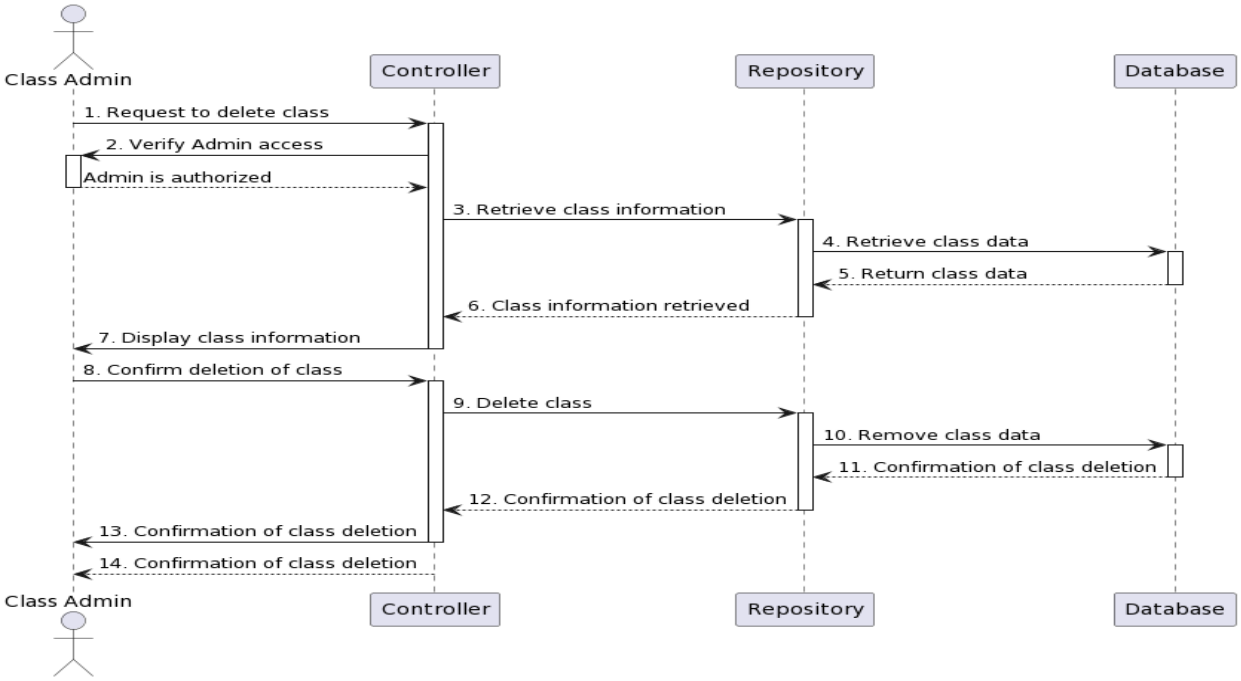


Figure 173 - <Super Admin, Class Admin> Delete Class

4.3.4.16. <Super Admin, Class Admin, Trainer> Export Class List

Summary: This diagram show how the Super Admin, Class Admin, Trainer delete class

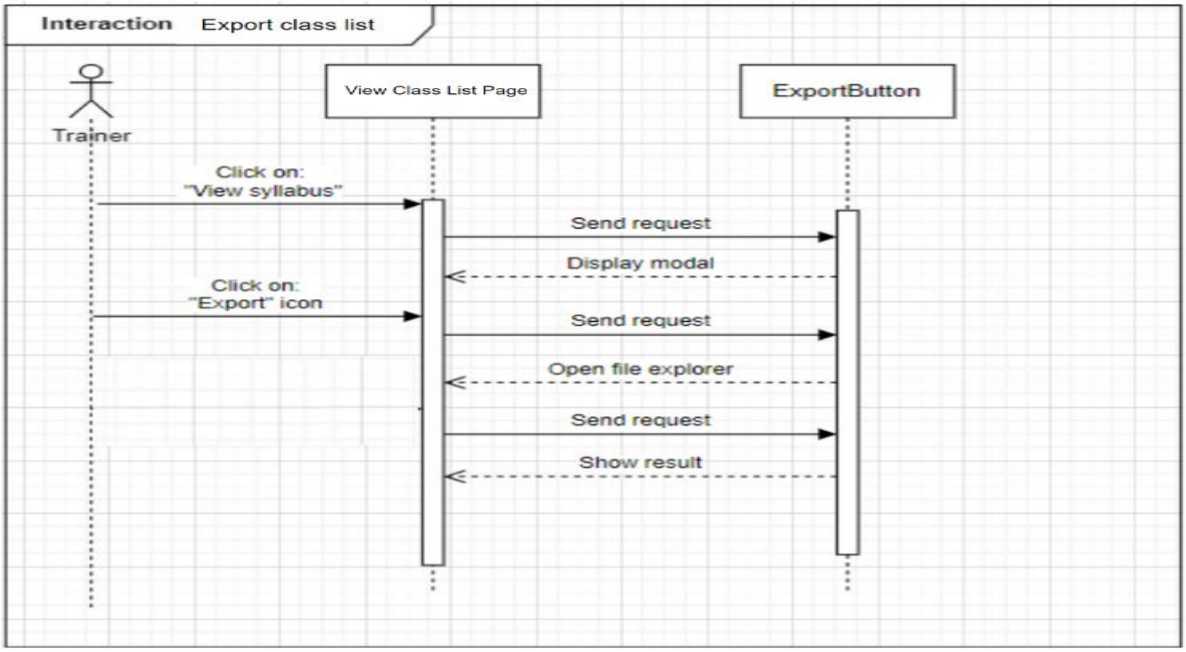


Figure 174 - <Super Admin, Class Admin, Trainer> Export Class List

4.3.5. Student Management Diagram

4.3.5.1. <Super Admin, Class Admin> List Student In System

Summary: This diagram show how Super Admin, Class Admin view list student in system

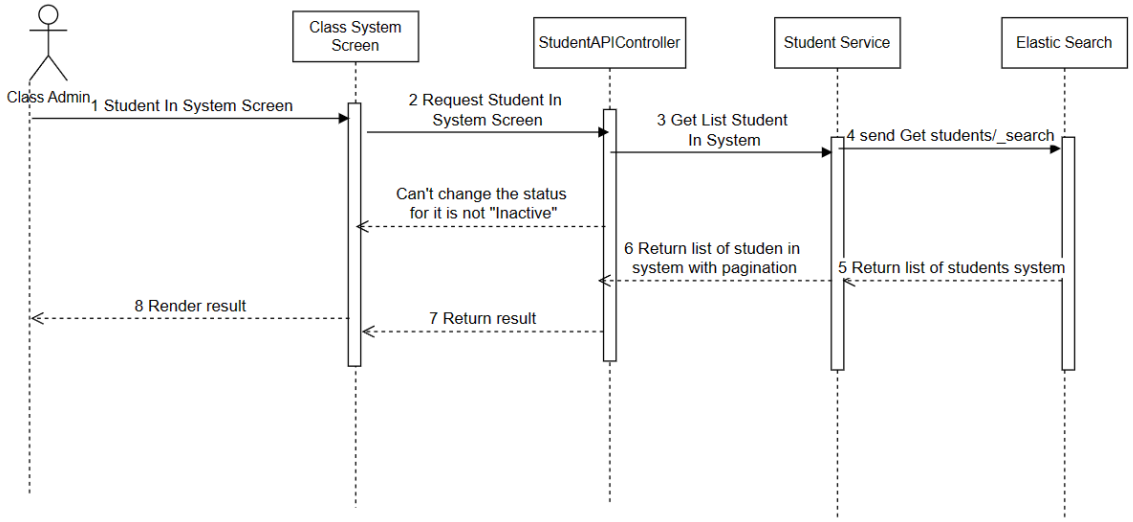


Figure 175 - <Super Admin, Class Admin> List Student In System

4.3.5.2. <Super Admin, Class Admin> List Student In Class

Summary: This diagram show how Super Admin, Class Admin view list student in class

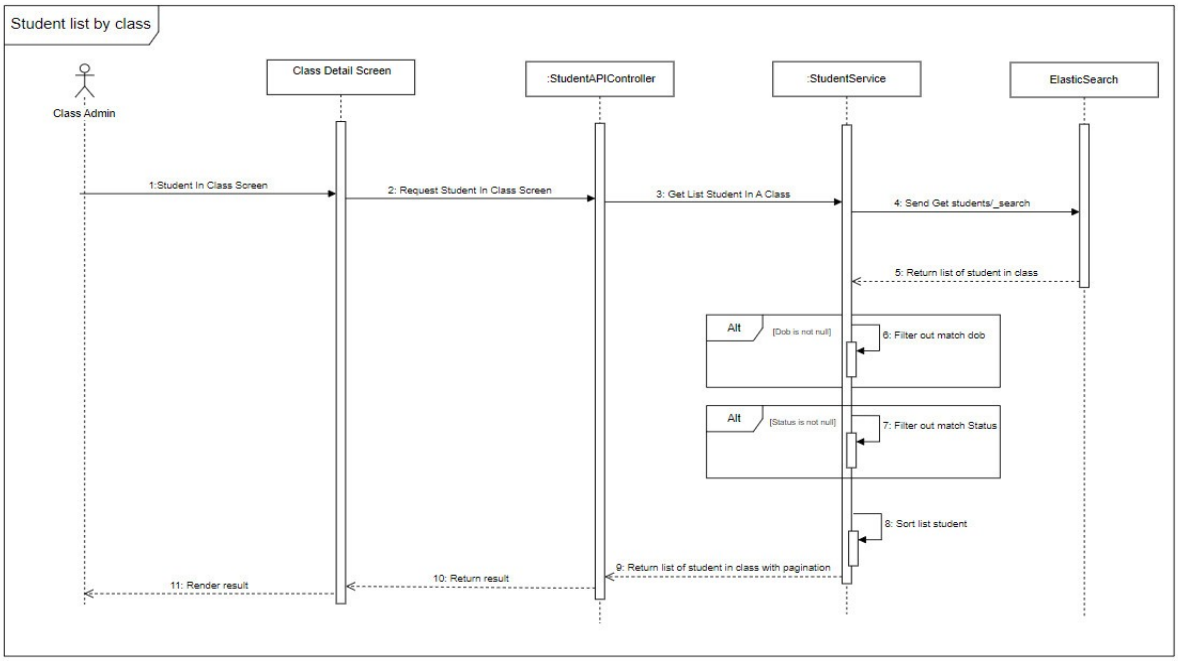


Figure 176 - <Super Admin, Class Admin> List Student In Class

4.3.5.3. <Super Admin, Class Admin> Edit Student Information Detail

Summary: This diagram show how Super Admin, Class Admin edit student information detail

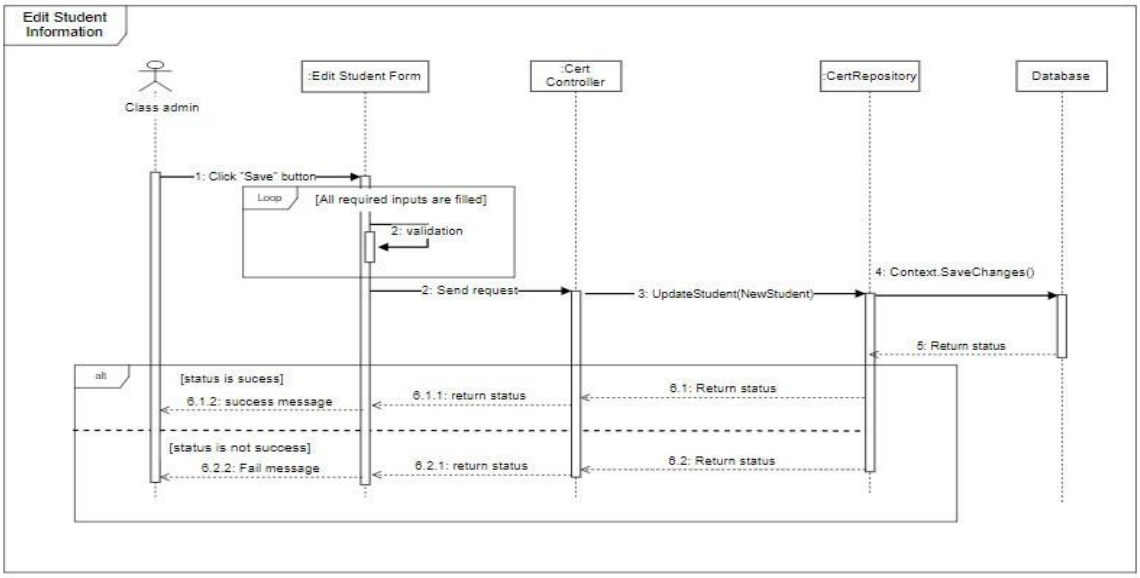


Figure 177 - <Super Admin, Class Admin> Edit Student Information

4.3.5.4. <Super Admin, Class Admin> Edit Student In Batch

Summary: This diagram show how Super Admin, Class Admin edit student in batch

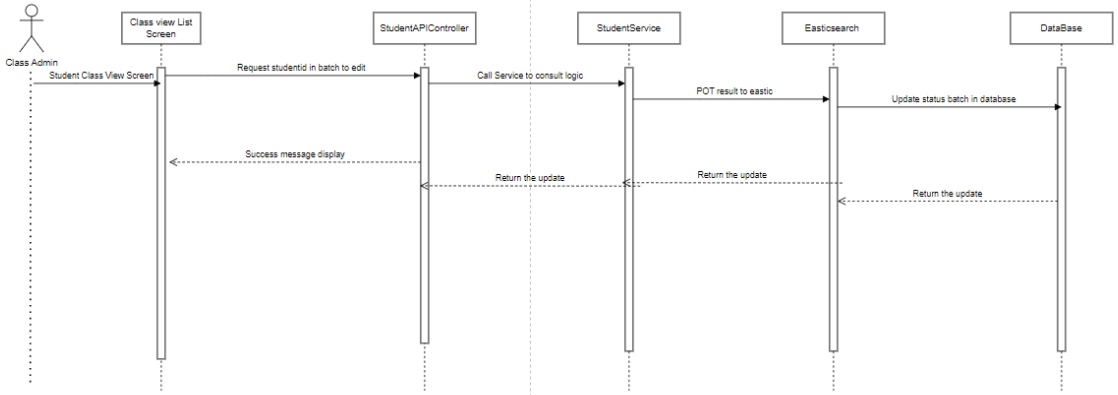


Figure 178 - <Super Admin, Class Admin> Edit Student In Batch

4.3.5.5. <Super Admin, Class Admin> Delete Student

Summary: This diagram show how Super Admin, Class Admin delete student

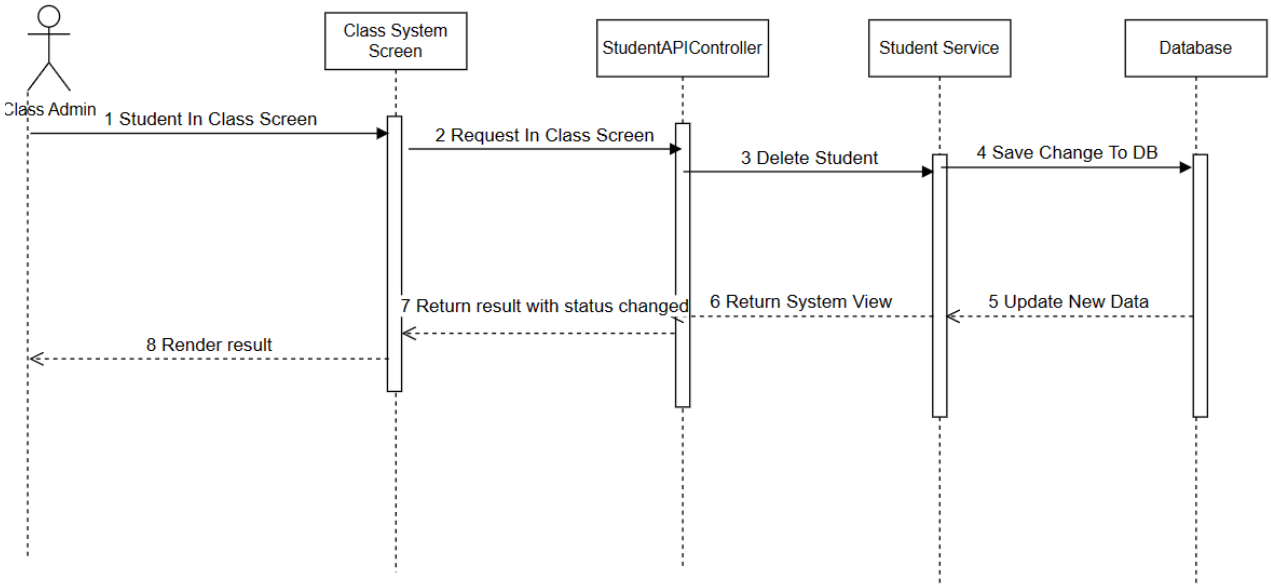


Figure 179 - <Super Admin, Class Admin> Delete Student

4.3.5.6. <Super Admin, Class Admin> Add New Student

Summary: This diagram show how Super Admin, Class Admin add new student

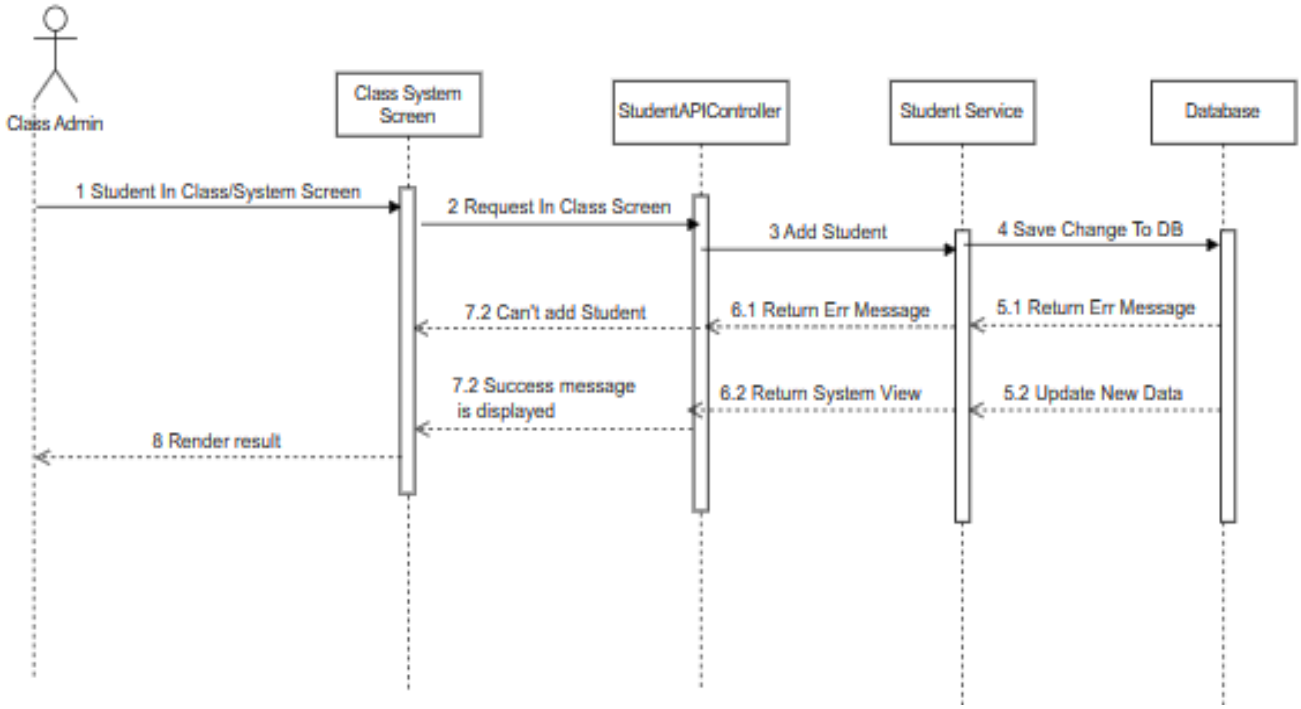


Figure 180 - <Super Admin, Class Admin> Add New Student

4.3.5.7. <Super Admin, Class Admin> Export Student List

Summary: This diagram show how Super Admin, Class Admin export student list

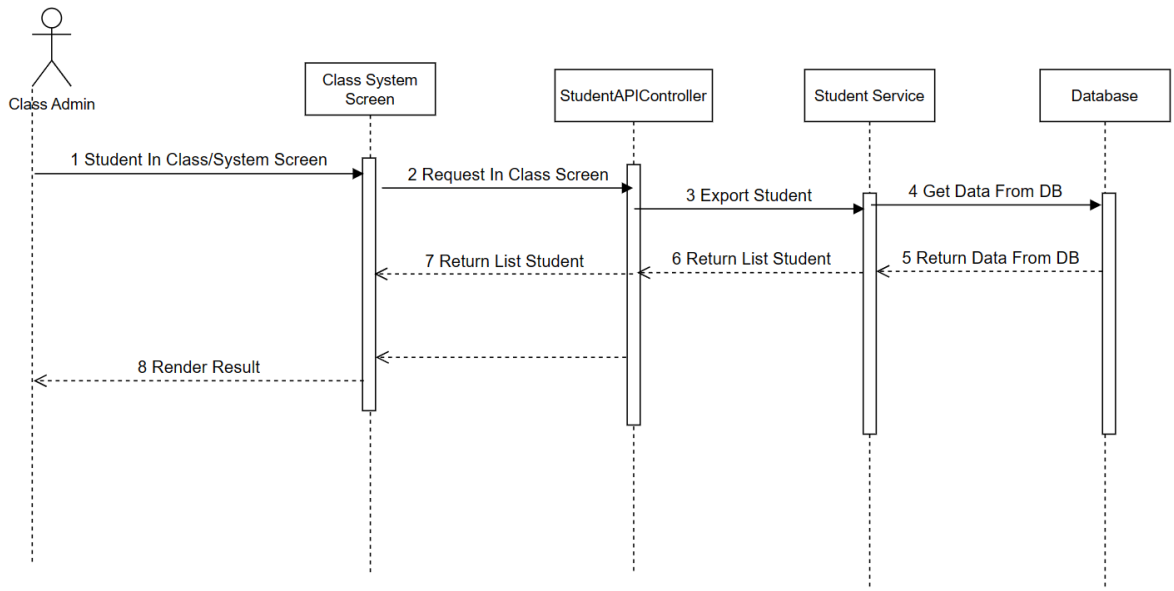


Figure 181 - <Super Admin, Class Admin> Export Student List

4.3.5.8. <Super Admin, Class Admin> Import Student List

Summary: This diagram show how Super Admin, Class Admin import student list

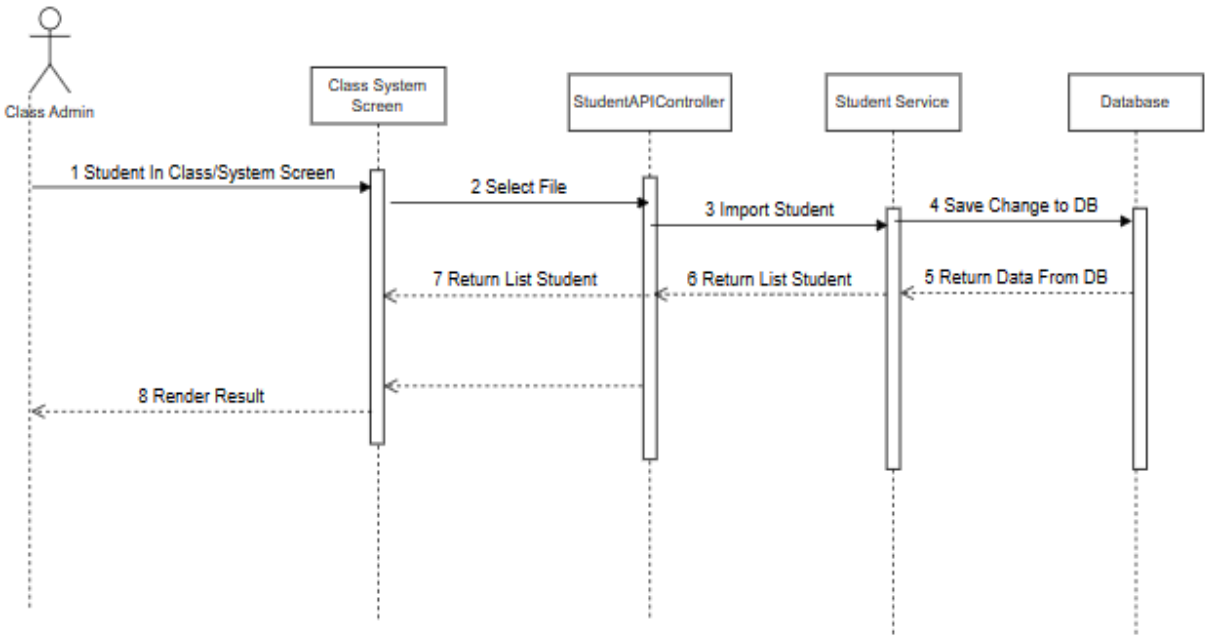


Figure 182 - <Super Admin, Class Admin> Import Student List

4.3.5.9. <Super Admin, Class Admin> Add New Class Information

Summary: This diagram show how Super Admin, Class Admin add new class information

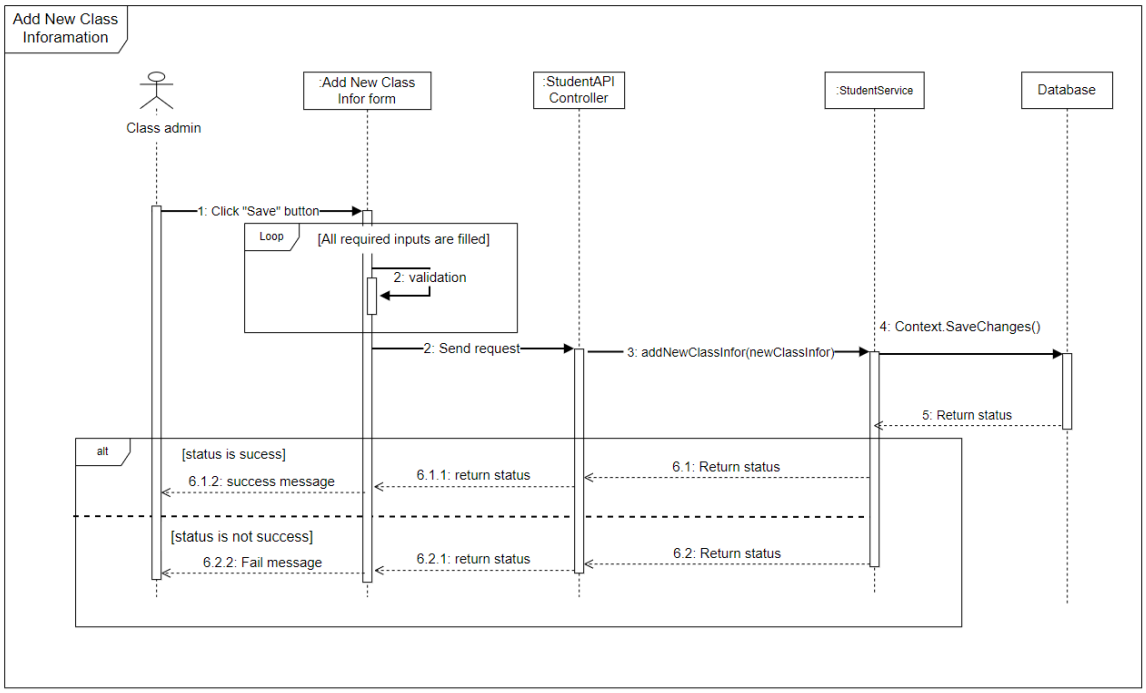


Figure 183 - <Super Admin, Class Admin> Add New Class Information

4.3.6. Score Management Diagram

4.3.6.1. <Super Admin, Class Admin, Trainer> Import list of trainee's score

Summary: This diagram show how Super Admin, Class Admin, Trainer import list of trainee's score

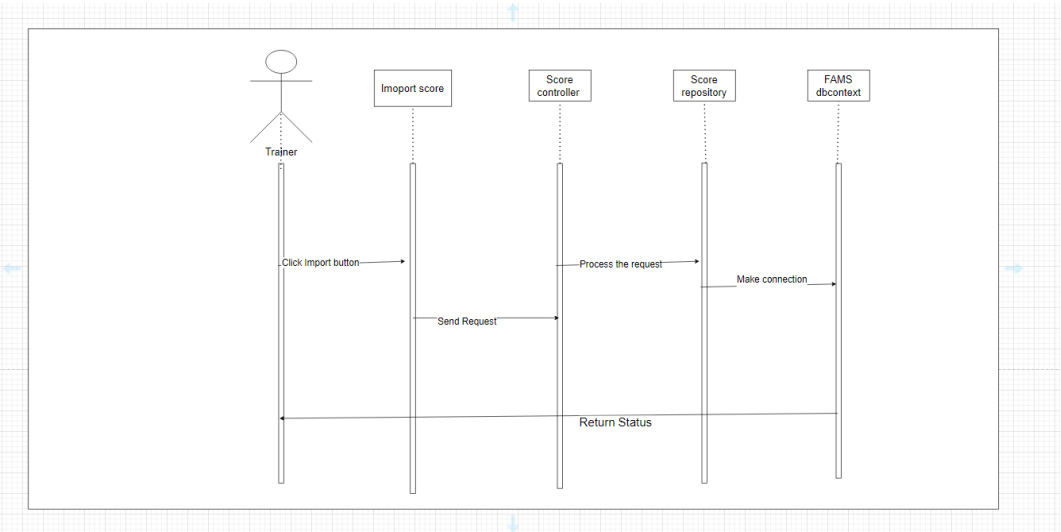


Figure 184 - <Super Admin, Class Admin, Trainer> Import list of trainee's score

4.3.6.2. <Super Admin, Class Admin, Trainer> Update the score list for each student

Summary: This diagram show how Super Admin, Class Admin, Trainer update the score list for each student

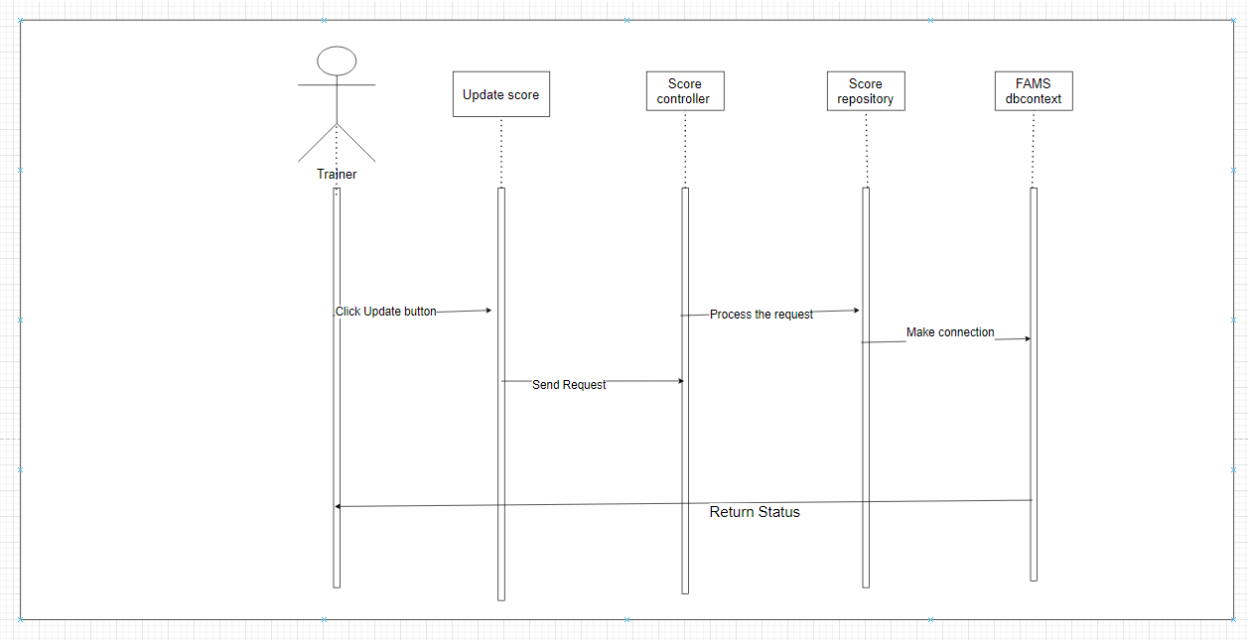


Figure 185 - <Super Admin, Class Admin, Trainer> Update the score list for each student

4.3.6.3. <Super Admin, Class Admin, Trainer> View the trainee's score, ranking and final assessment of all modules per class

Summary: This diagram show how Super Admin, Class Admin, Trainer view the trainee's score, ranking and final assessment of all modules per class

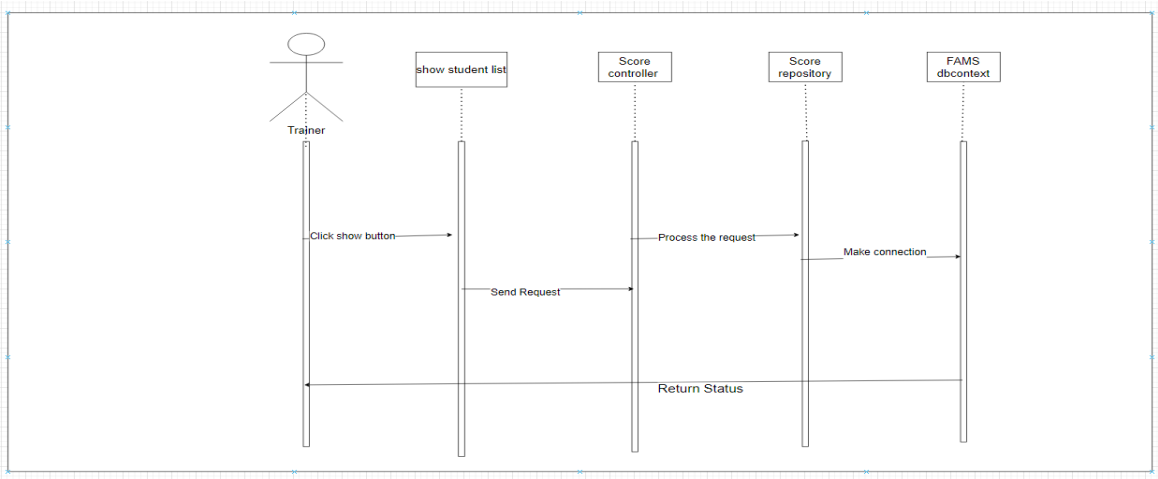


Figure 186 - <Super Admin, Class Admin, Trainer> View the trainee's score, ranking and final assessment of all modules per class

4.3.6.4. <Super Admin, Class Admin, Trainer> View final studying result per trainee

Summary: This diagram show how Super Admin, Class Admin, Trainer view final studying result per trainee

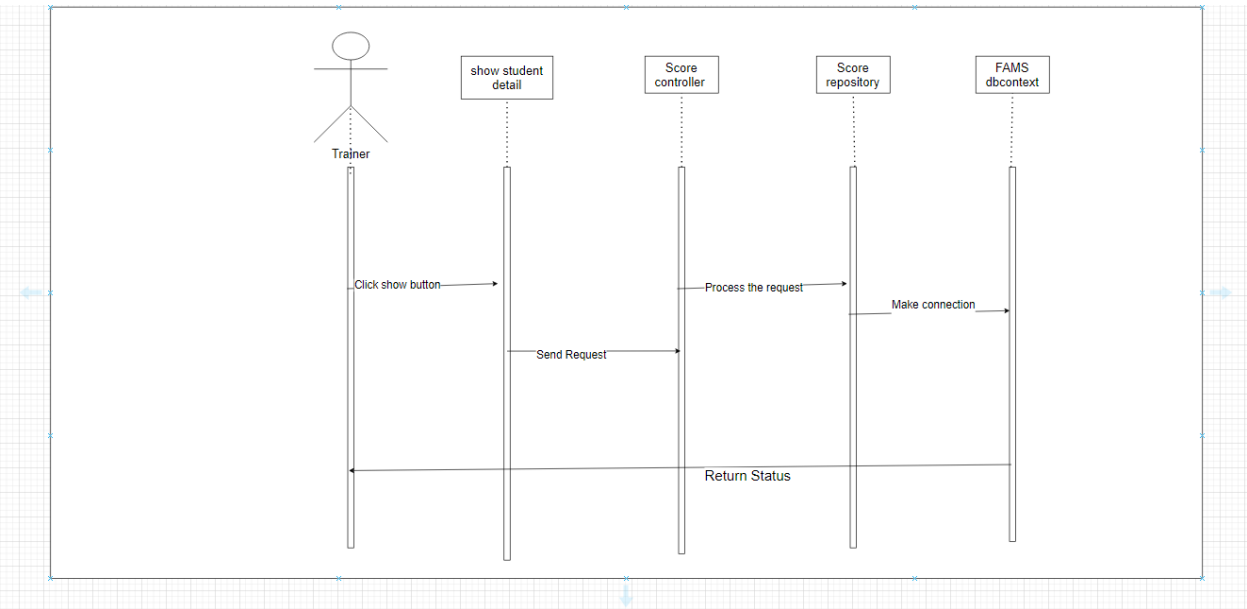


Figure 187 - <Super Admin, Class Admin, Trainer> View final studying result per trainee

4.3.6.5. <Super Admin, Class Admin, Trainer> Update status of certificate award

Summary: This diagram show how Super Admin, Class Admin, Trainer update status of certificate award

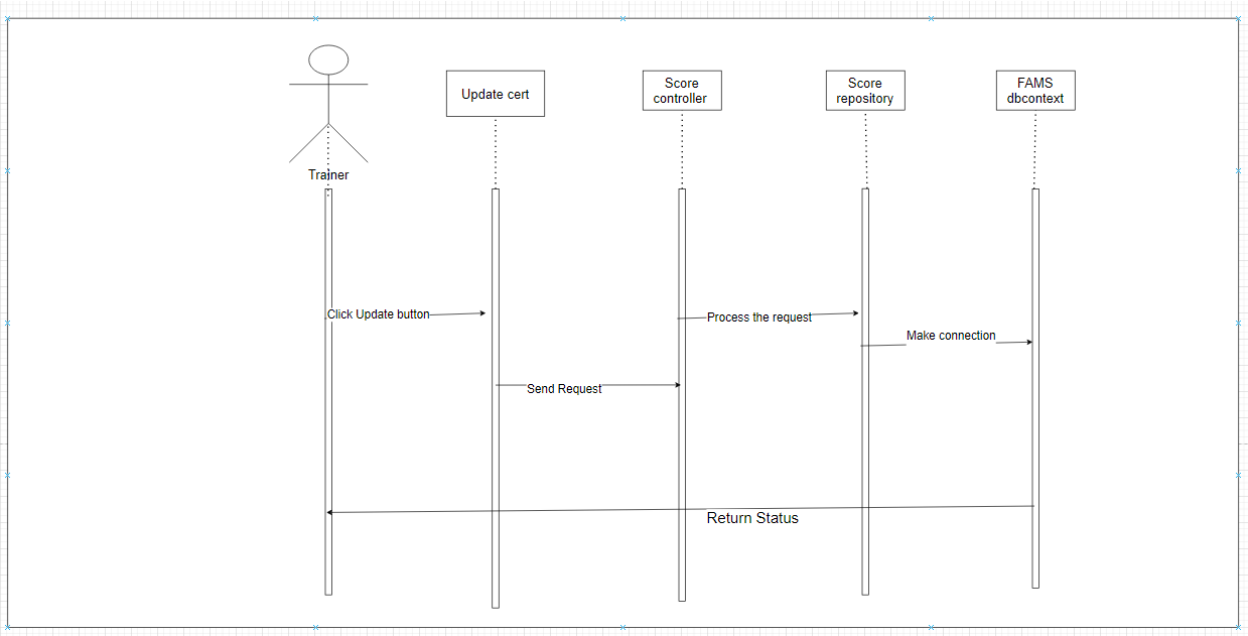


Figure 188 - <Super Admin, Class Admin, Trainer> Update status of certificate award

4.3.6.6. <Super Admin, Class Admin, Trainer> Export Student Score List

Summary: This diagram show how Super Admin, Class Admin, Trainer export student score list

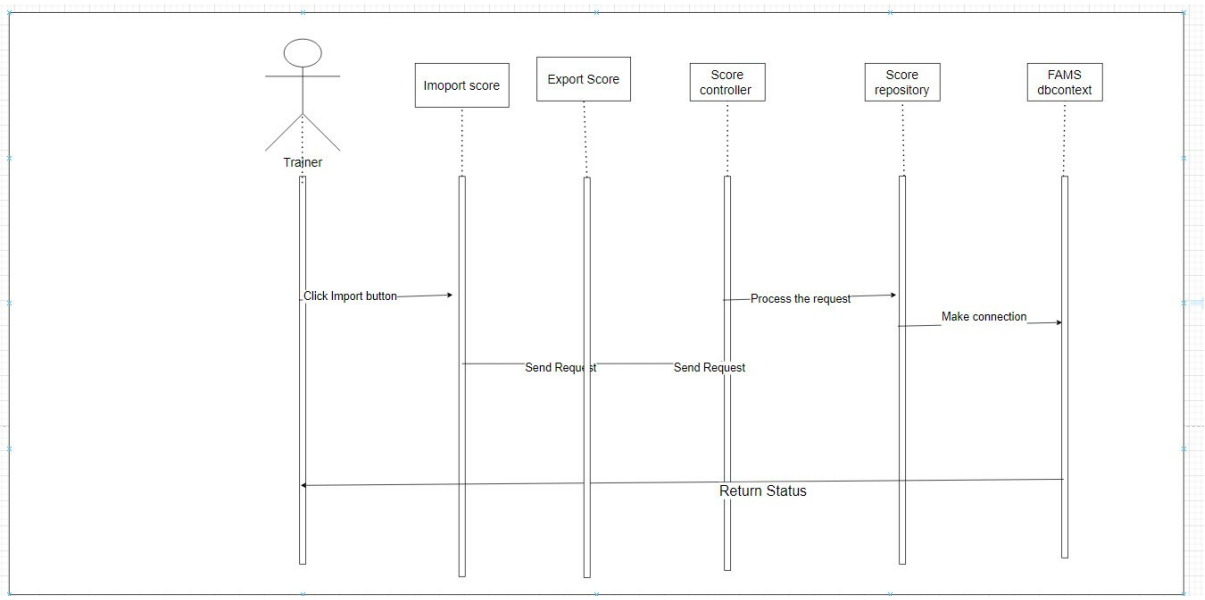


Figure 189 - <Super Admin, Class Admin, Trainer> Export export student score list

4.3.7. Reservation Management Diagram

4.3.7.1. <Super Admin, Class Admin> View list of students with status "Reservation"

Summary: This diagram show how Super Admin, Class Admin view list of students with status "Reservation"

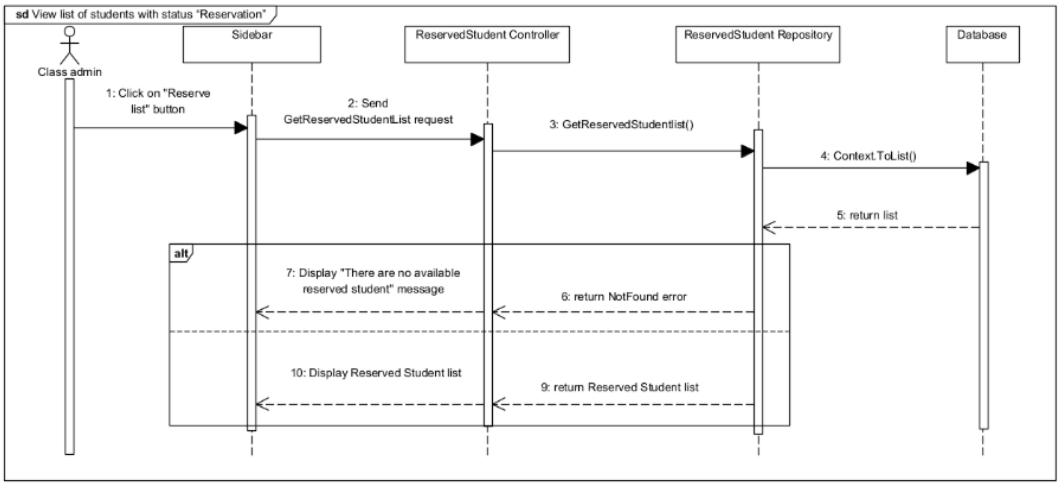


Figure 190 - <Super Admin, Class Admin> View list of students with status "Reservation"

4.3.7.2. <Super Admin, Class Admin> Update list of students with status "Reservation"

Summary: This diagram show how Super Admin, Class Admin update list of students with status "Reservation"

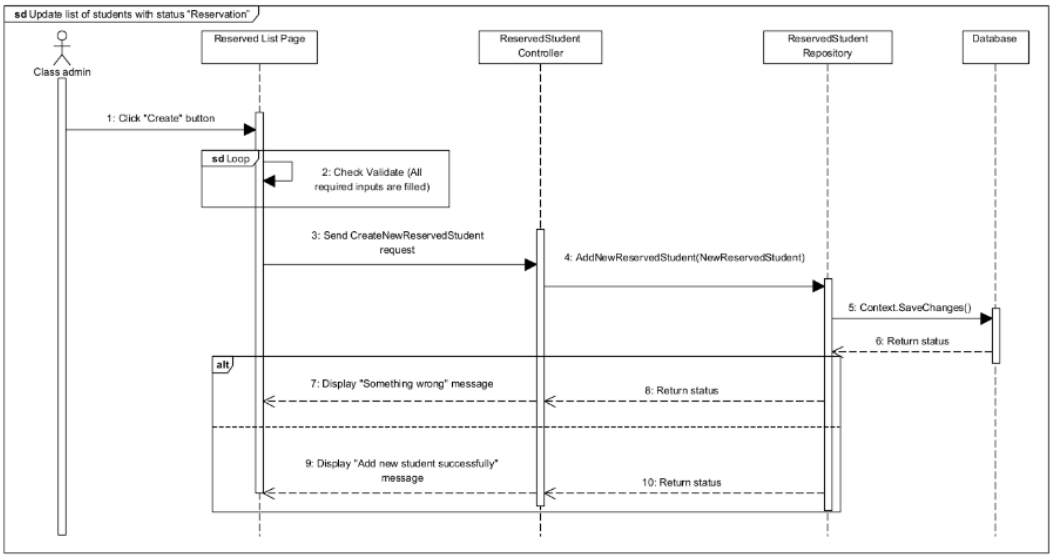


Figure 191 - <Super Admin, Class Admin> Update list of students with status "Reservation"

4.3.7.3. Super Admin, Class Admin> Terminate/Remove status "Reservation"

Summary: This diagram show how Super Admin, Class Admin terminate/remove status "Reservation"

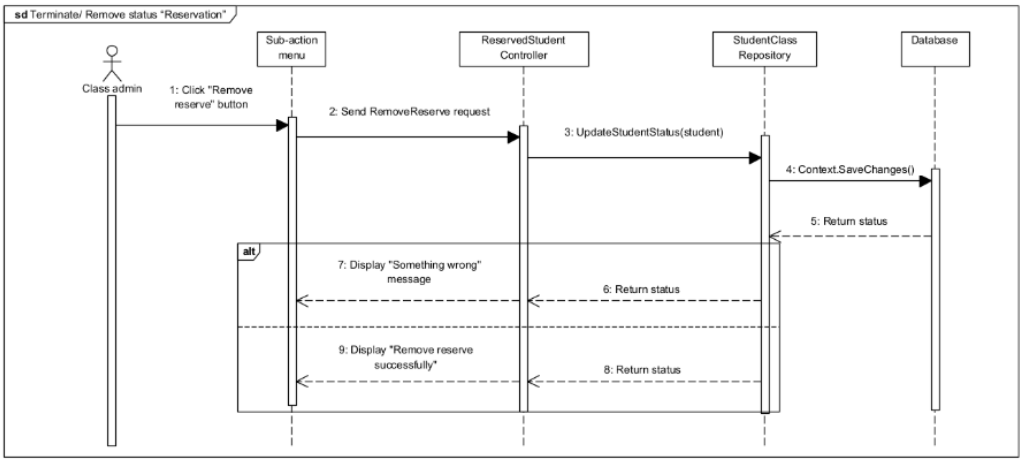


Figure 192 - <Super Admin, Class Admin> Terminate/Remove status "Reservation"

4.3.7.4. <Super Admin, Class Admin> Find new class for student after "Reservation"

Summary: This diagram show how Super Admin, Class Admin find new class for student after "Reservation"

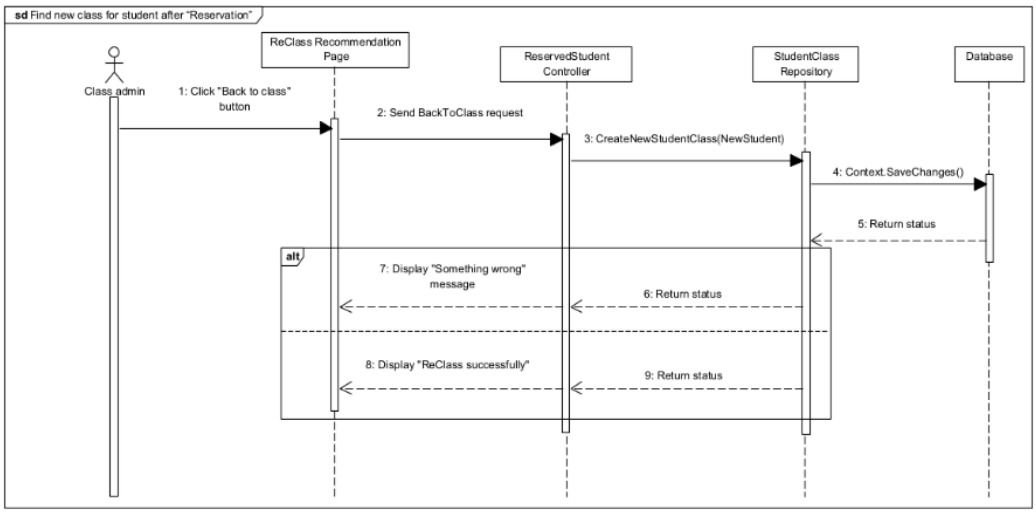


Figure 193 - <Super Admin, Class Admin> Find new class for student after "Reservation"

4.3.7.5. <Super Admin, Class Admin>Update status "Drop out" for student

Summary: This diagram show how Super Admin, Class Admin update status "Drop out" for student

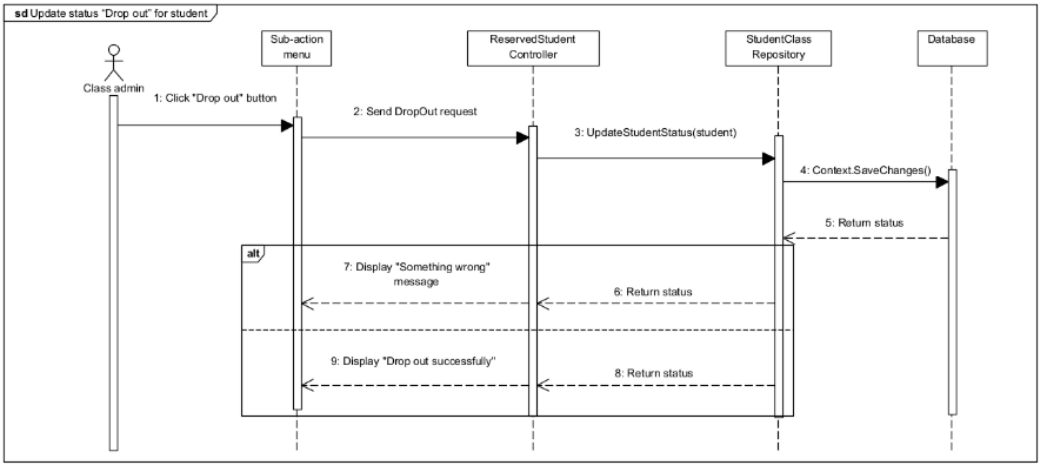


Figure 194 - <Super Admin, Class Admin> Update status "Drop out" for student

4.3.7.6. <Super Admin, Class Admin> Export Reserve List

Summary: This diagram show how Super Admin, Class Admin export reserve list

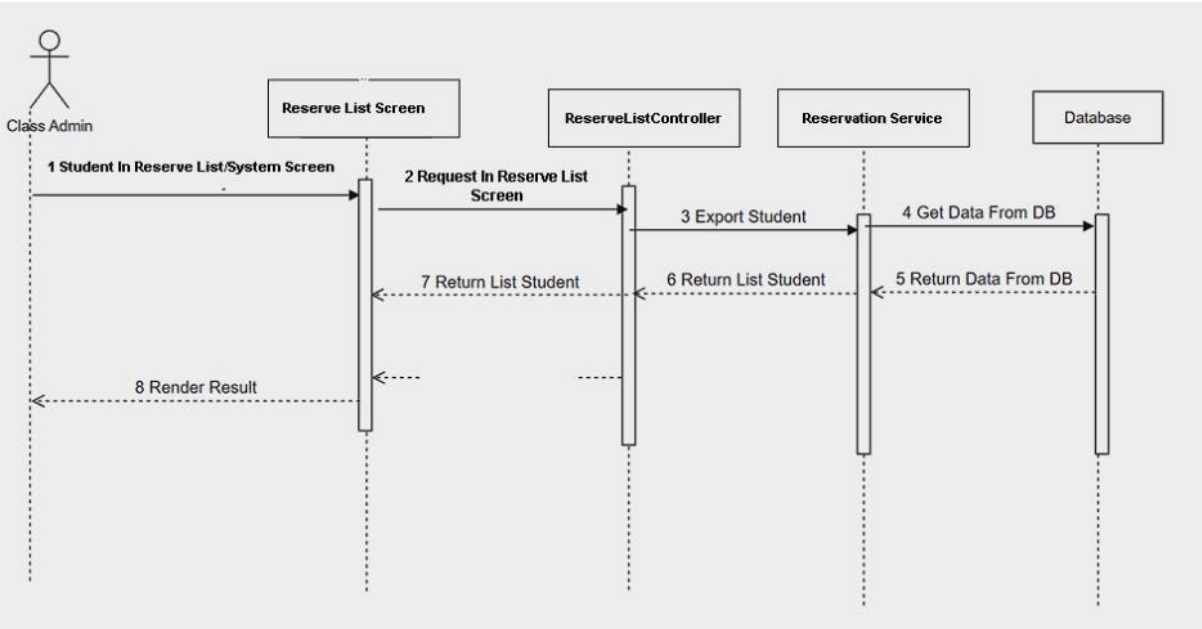


Figure 195 - <Super Admin, Class Admin> Export Reserve List

4.3.7.7. <Super Admin, Class Admin> Import Reserve List

Summary: This diagram show how Super Admin and Class Admin import reserve list

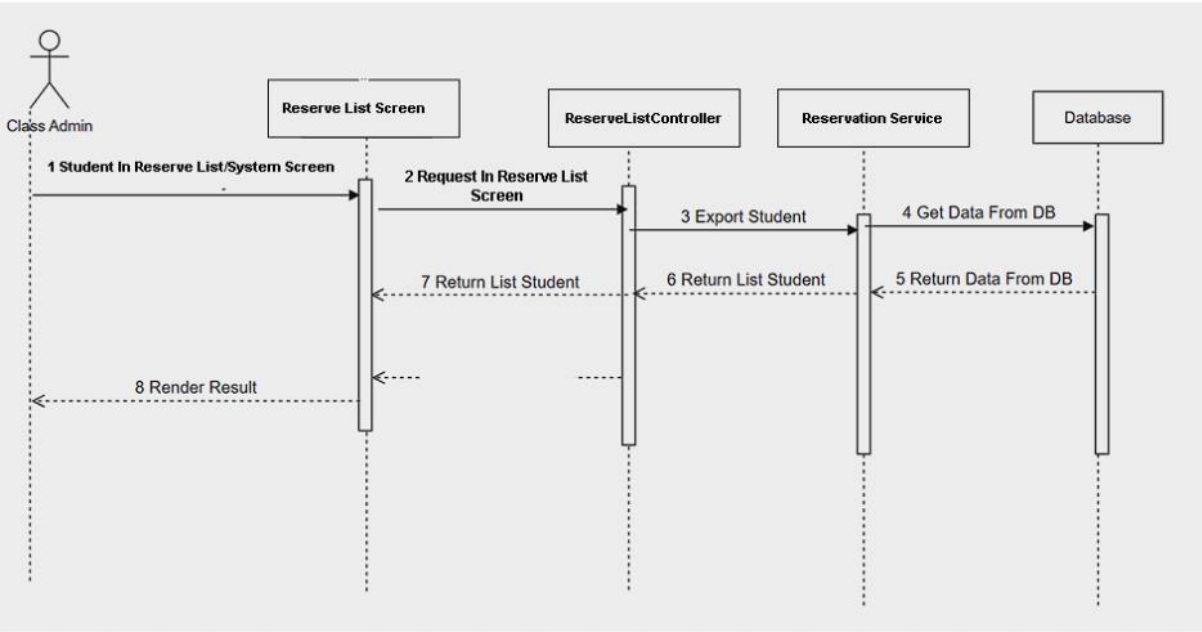


Figure 196 - <Super Admin, Class Admin> Import Reserve List

4.3.8. Email Inform/Remind Diagram

4.3.8.1. <Super Admin, Class Admin> Add Email Template

Summary: This diagram show how Super Admin, Class Admin add email template

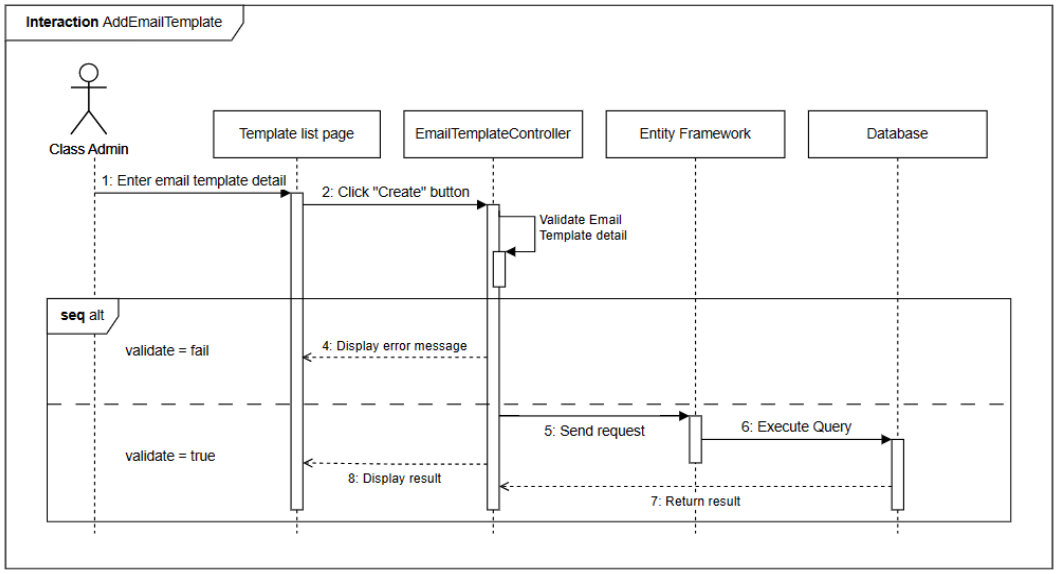


Figure 197 - <Super Admin, Class Admin> Add Email Template

4.3.8.2. <Super Admin, Class Admin> Search Email Template

Summary: This diagram show how Super Admin, Class Admin search email template

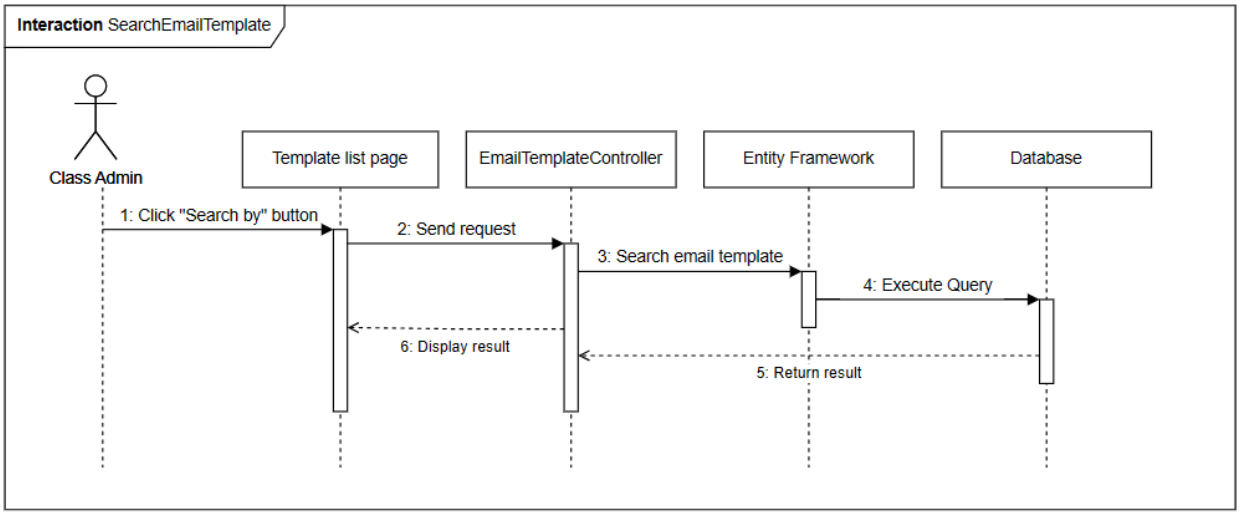


Figure 198 - <Super Admin, Class Admin> Search Email Template

4.3.8.3. <Super Admin, Class Admin> Edit Email Template

Summary: This diagram show how Super Admin, Class Admin edit email template

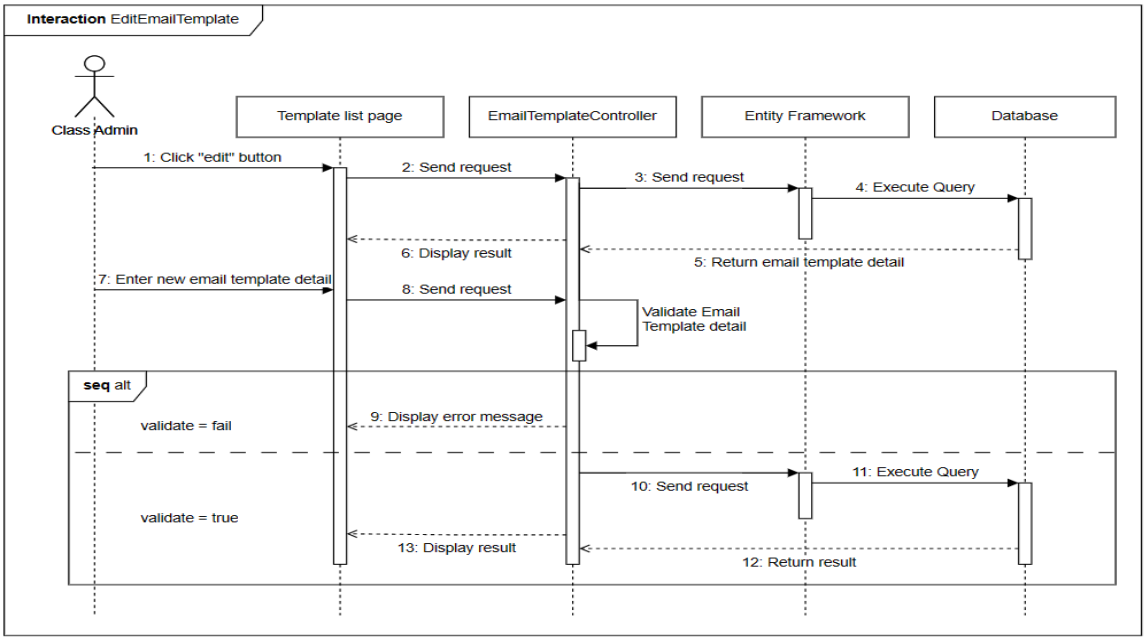


Figure 199 - <Super Admin, Class Admin> Edit Email Template

4.3.8.4. <Super Admin, Class Admin> Send Email To Trainer

Summary: This diagram show how Super Admin, Class Admin send email to trainer

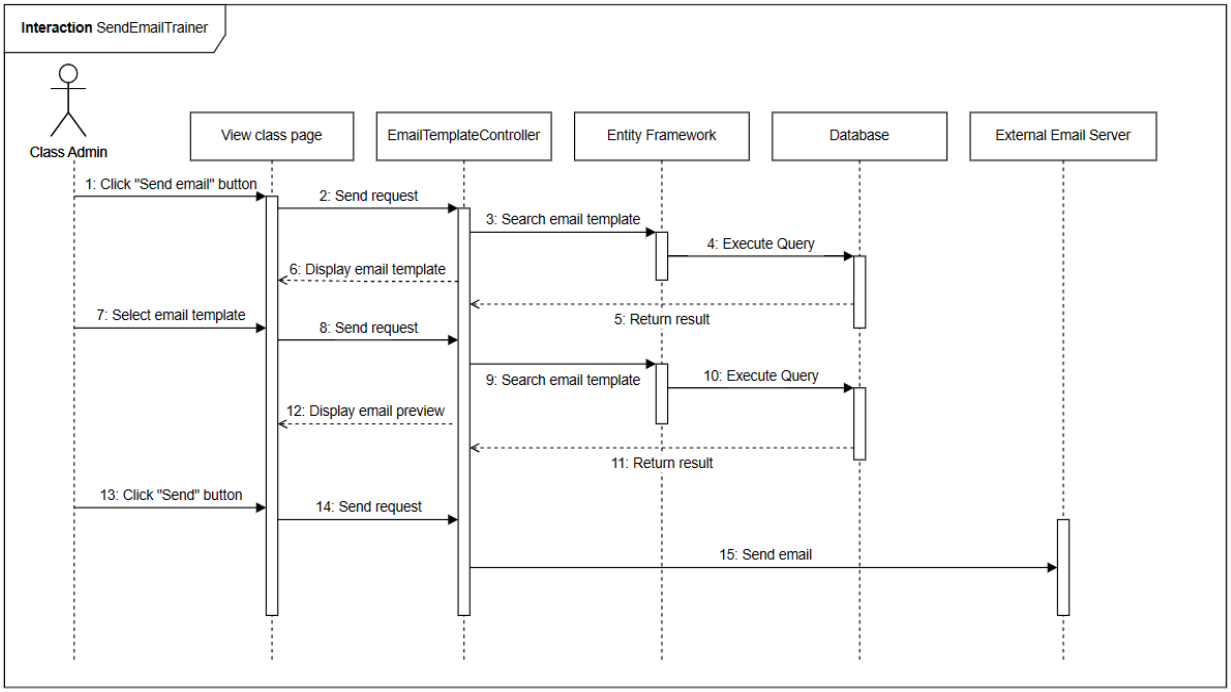


Figure 200 - <Super Admin, Class Admin> Send Email To Trainer

4.3.8.5. <Super Admin, Class Admin> Send Email To Reserve

Summary: This diagram show how Super Admin, Class Admin send email to reserve

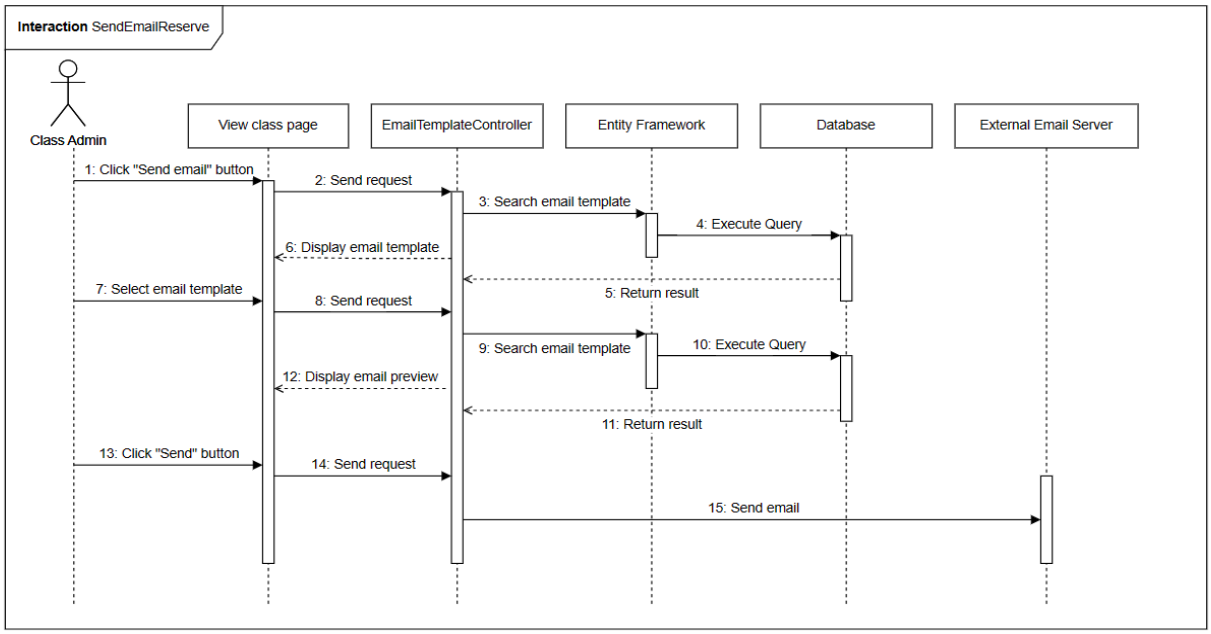


Figure 201 - <Super Admin, Class Admin> Send Email To Reserved Student

4.3.8.6. <Super Admin, Class Admin> Send Email To Student

Summary: This diagram show how Super Admin, Class Admin send email to student

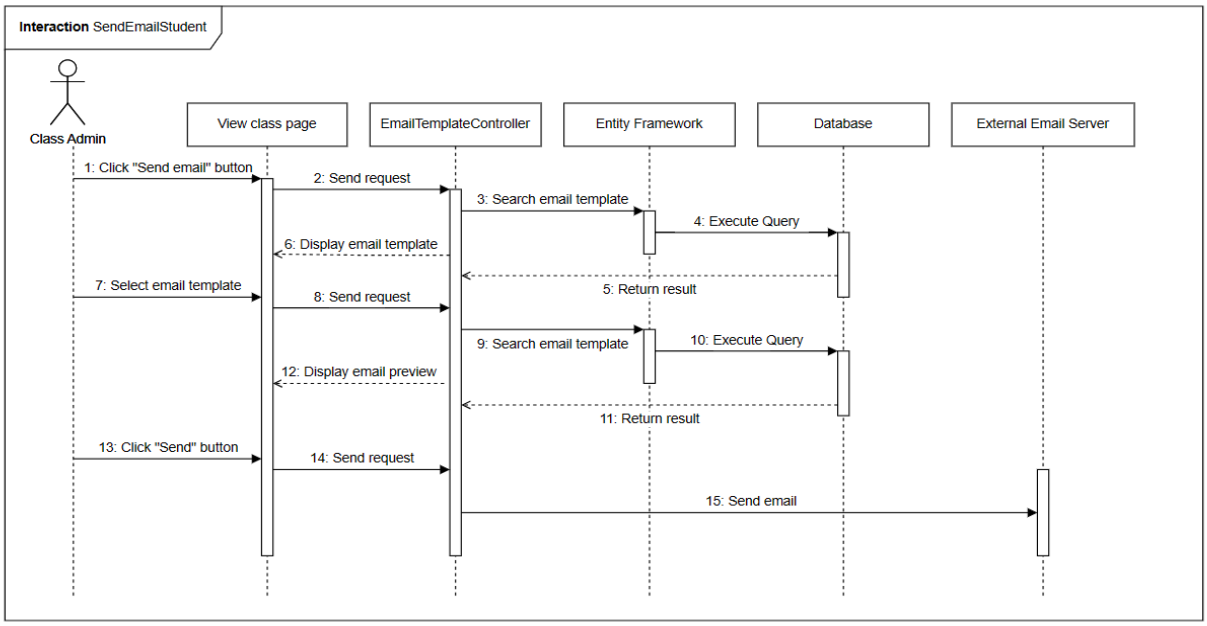


Figure 202 - <Super Admin, Class Admin> Send Email To Student

4.3.8.7. <Super admin, Class admin> Export Email Template list

Summary: This diagram show how Super Admin, Class Admin export email template list

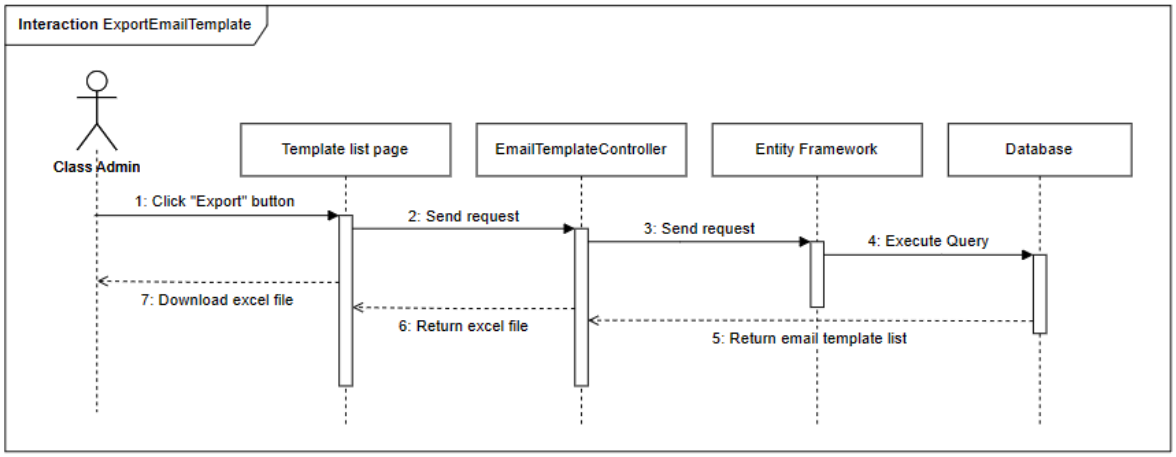


Figure 203 - <Super admin, Class admin> Export Email Template List

5. User Interface Design

5.1. User Management Interface

5.1.1. Login Page

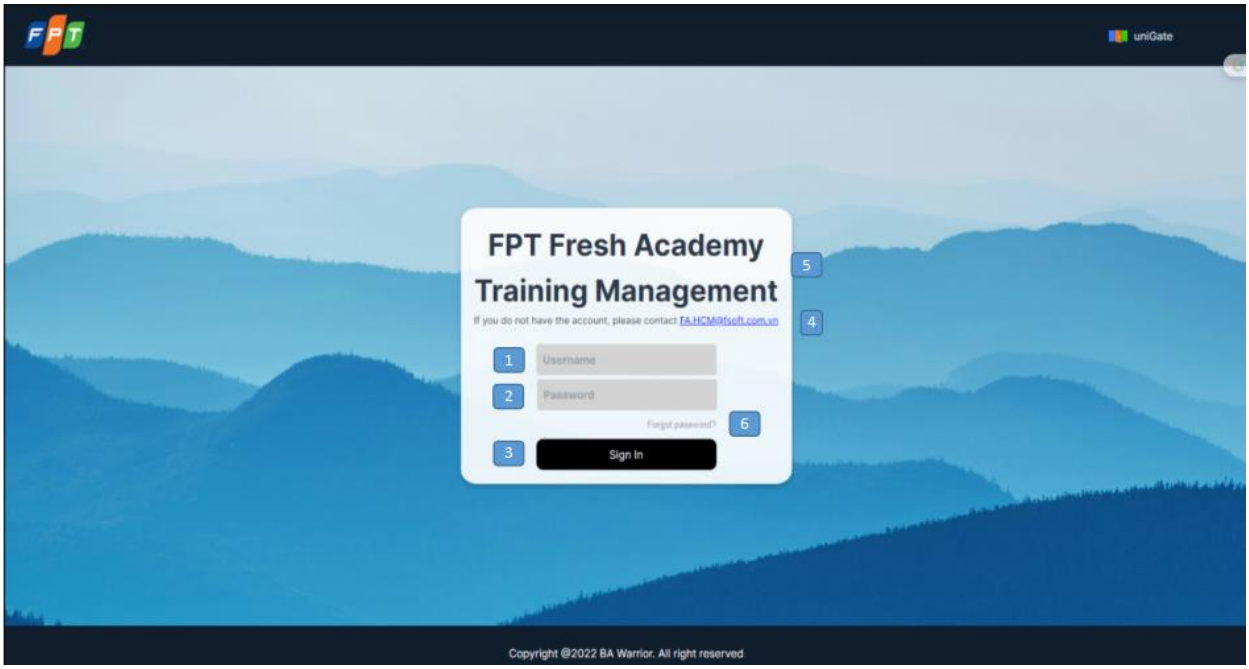


Figure 204 - Login Page UI 1

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Username	Username of user	No	Yes	Textbox	String
2	Password	Password of user	No	Yes	Textbox	String
5	Login Background	A back ground	Yes	No	Image	Image

Table 12 - <User Interface-Field> Login Page

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome

3	Sign in	Click to sign in	Validate all required fields	Go to main menu
4	Contact link	Link that help new user contact with admin	N/A	Go back to Login Page
6	Forgot password?	Click to redirect to forgot password page	N/A	Go to forgot password page

Table 13 - <User Interface-Button/Hyperlinks> Login Page

5.1.2. Forgot Password Page

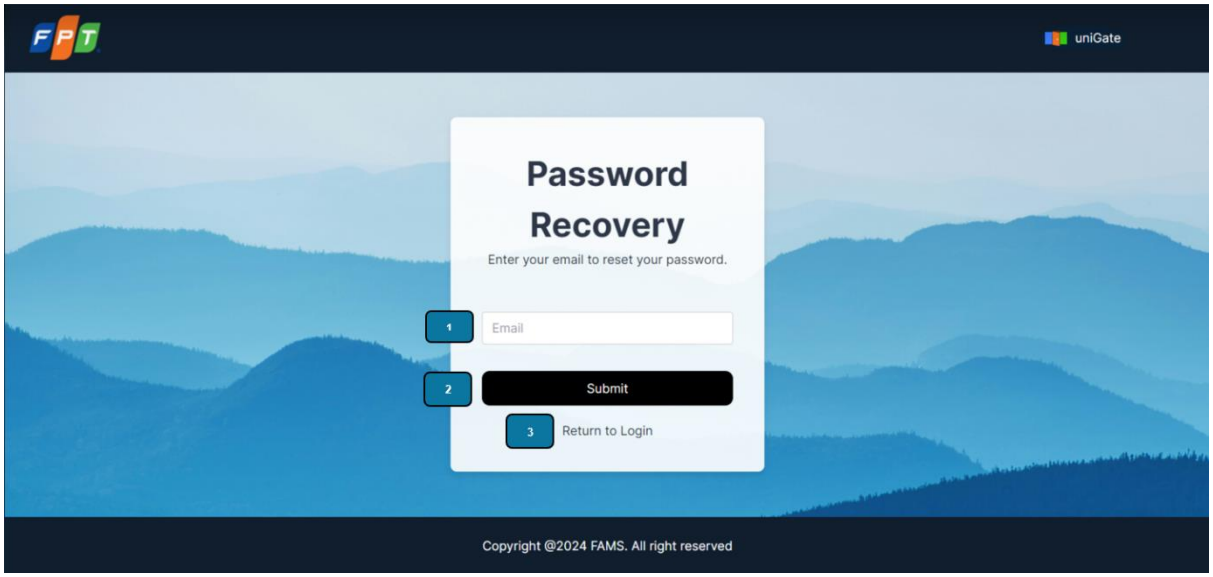


Figure 205 – Forgot Password Page 1

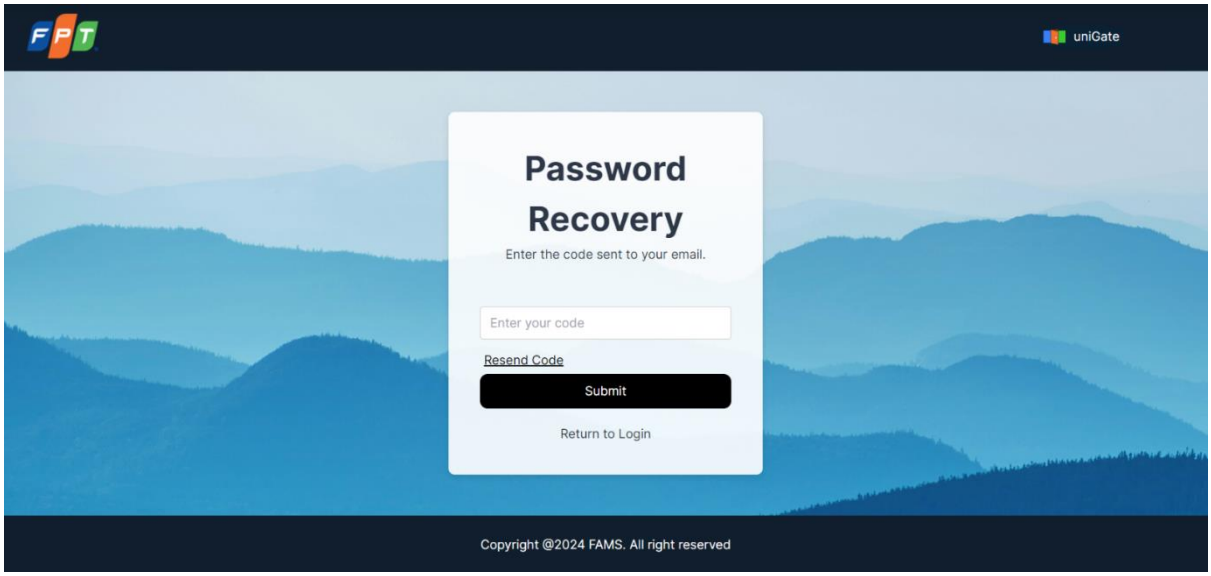


Figure 206 – Forgot Password Page 2

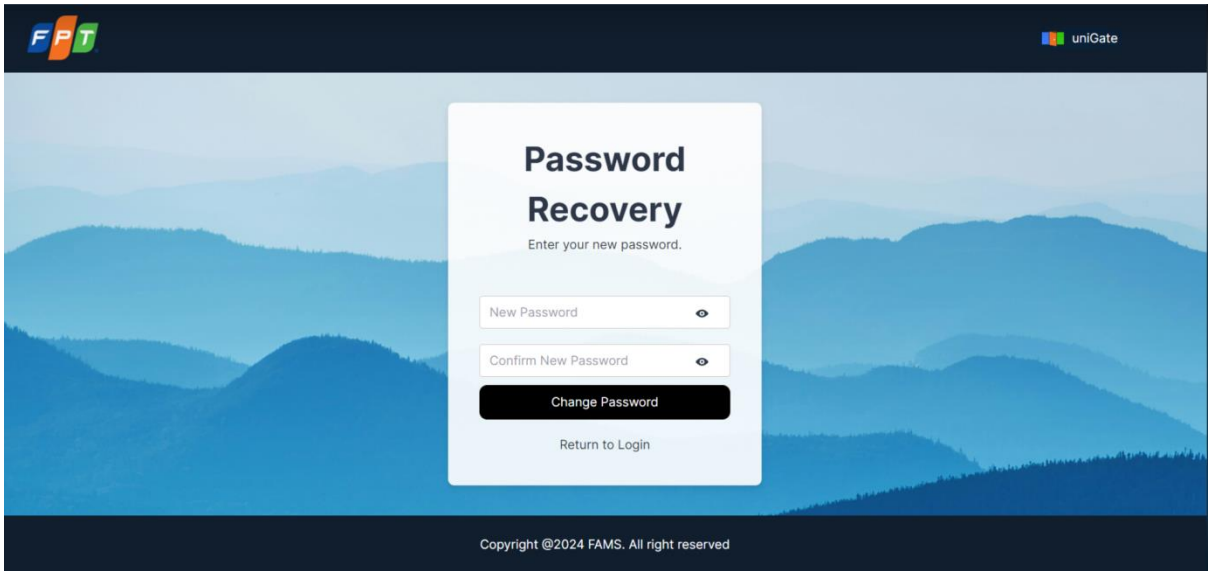


Figure 207 - Forgot Password Page 3

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Email	Email of user	No	Yes	Textbox	String
4	Enter your code	Code send to recover passwod	No	Yes	Textbox	String
7	New Password	New Password for	No	Yes	Textbox	String

		user account				
8	Confirm New Password	Confirm New Password for user account	No	Yes	Textbox	String

Table 14 - <User Interface-Field> Forgot Password Page

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
2	Submit	Click to reset password	Validate password fields	Go to recover password page
3	Return to login	Click to return login page	N/A	Go to login page
6	Submit	Click to redirect to recover password page	- Validate code fields - Code send must be valid	Go to recover password page
9	Change Password	Click to recover password	- Validate password fields - "New password" and "Confirm new password" must be the same	Go to login page

Table 15 - <User Interface-Field> Forgot Password Page

5.1.3. User Management Page

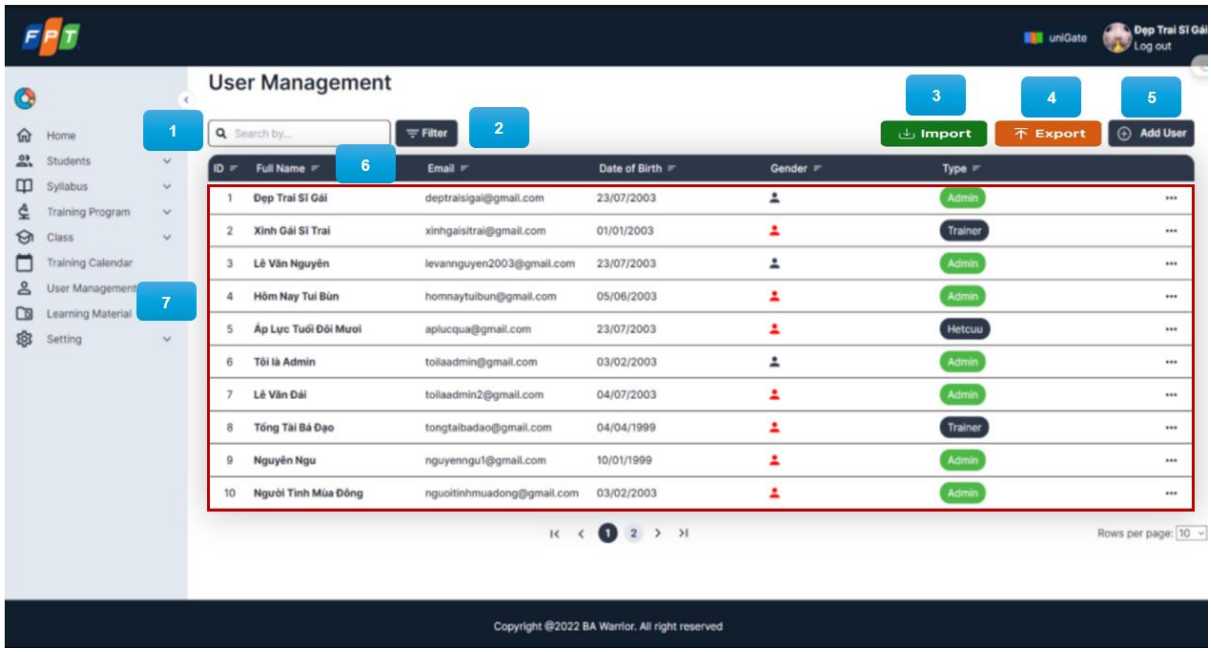


Figure 208 - User Management Page UI 1

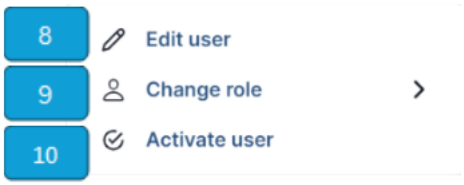


Figure 209 - Edit Menu - User Management Page UI 2

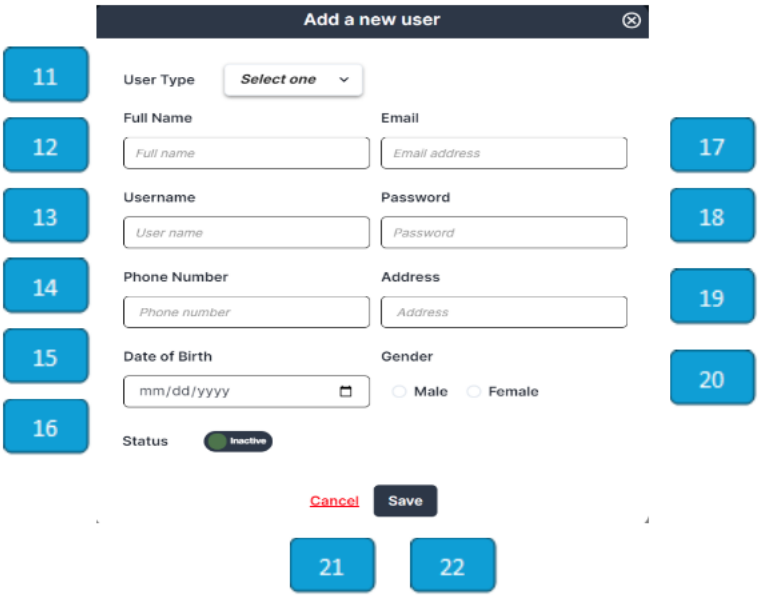


Figure 210 - Add User Pop Up - User Management Page UI 3

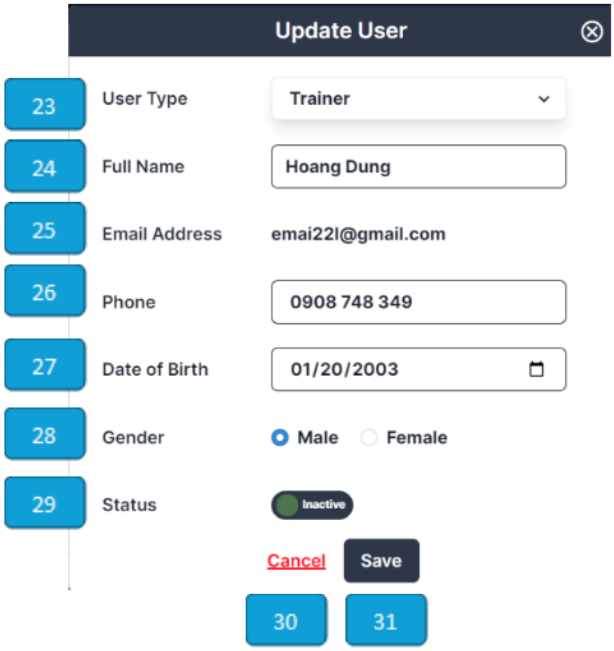


Figure 211 - Update User Pop Up - User Management Page UI 4

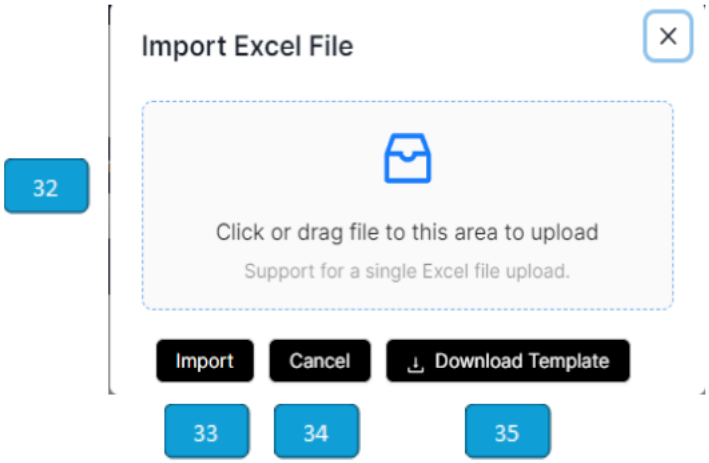


Figure 212 - Import User Pop Up - User Management Page UI 5

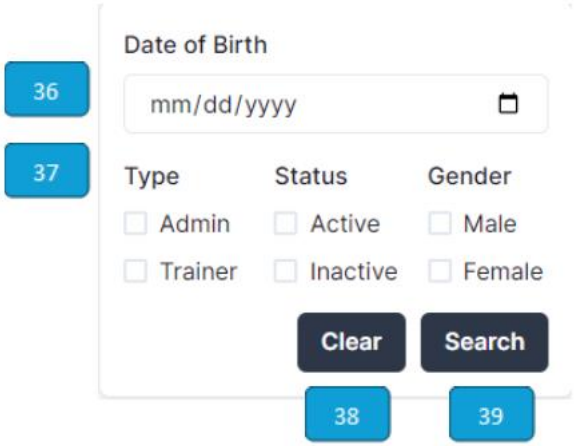


Figure 213 - Filter Menu - User Management Page UI 6

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Search	Search User	No	Yes	Textbox	String
6	List User	Show User information	Yes	Yes	Table	Table Data
11	User Type	Choose user role	No	Yes	Dropdown box	String
12	Full name	Full name of user	No	Yes	Textbox	String
13	Username	Username of user	No	Yes	Textbox	String
14	Phone number	Phone number of user	No	Yes	Textbox	String
15	Date of birth	Date of birth	No	Yes	Datetime picker	datetime
16	Status	Status of user	No	Yes	Toggle button	String
17	Email address	Email of user	No	Yes	Textbox	String
18	Password	Password of user	No	Yes	Textbox	String

19	Address	Address of user	No	Yes	Textbox	String
20	Gender	Gender of user	No	Yes	Radio button	String
23	User Type	Choose user role	No	Yes	Dropdown box	String
24	Full name	Full name of user	No	Yes	Textbox	String
25	Email address	Email of user	Yes	Yes	Label	String
26	Phone number	Phone number of user	No	Yes	Textbox	String
27	Date of birth	Date of birth	No	Yes	Datetime picker	datetime
28	Gender	Gender of user	No	Yes	Radio button	String
29	Status	Status of user	No	Yes	Toggle button	String
32	Upload area	Area to upload excel file to import	No	Yes	Drag and drop area	Excel file
36	Date of birth	Date of birth	No	Yes	Datetime picker	datetime
37	Filter Option	Choose type, status, gender of user	No	Yes	Radio button	String

Table 16 - <User Interface-Field> User Management Page

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Filter	Click to open Filter User menu	N/A	Go to filter menu
3	Import	Click to open Import User Pop up	N/A	Go to Import User Pop Up
4	Export	Click to Export File of User List	N/A	Exported user list data in the format Excel

5	Add User	Click to open Add User Pop up	N/A	Go to Add User Pop Up
6	Sort User	Click to Soft User	N/A	Sorted User show on List User
7	Edit Menu	Click to open Edit Menu	N/A	Go to Edit User Menu
8	Edit User	Click to open Update User Pop up	N/A	Go to Update User Pop Up
9	Change User's role	Click to Change User's Role	N/A	Sent request to system to change user's role
10	Change User's status	Click to Edit User	N/A	Sent request to system to change user's status
21	Cancel add user	Click to Cancel Add User	N/A	Close pop up Go to User Management Page
22	Save a new user	Click to Save A New User	N/A	Close pop up Sent request to system to Add User Go to User Management Page
30	Cancel update user	Click to Cancel Add User	N/A	Close pop up Go to User Management Page
31	Save update user	Click to Save A New User	N/A	Close pop up Sent request to system to Add User Go to User Management Page
33	Import	Click to Import User from excel file	N/A	Sent request to system to Import User Show message when success import Show all wrong verify when fails import
34	Cancel	Click to cancel Import User	N/A	Close pop up Go to User Management Page

35	Download template	Click to download template	N/A	Download ImportUserTemplate.xlsx
38	Clean	Click to clean all option in filter menu	N/A	Clean all option
39	Search	Click to start filer	N/A	Close pop up Go to User Management Page with filtered user list

Table 17 - <User Interface-Button/Hyperlinks> User Management Page

5.1.4. User's Permission Page

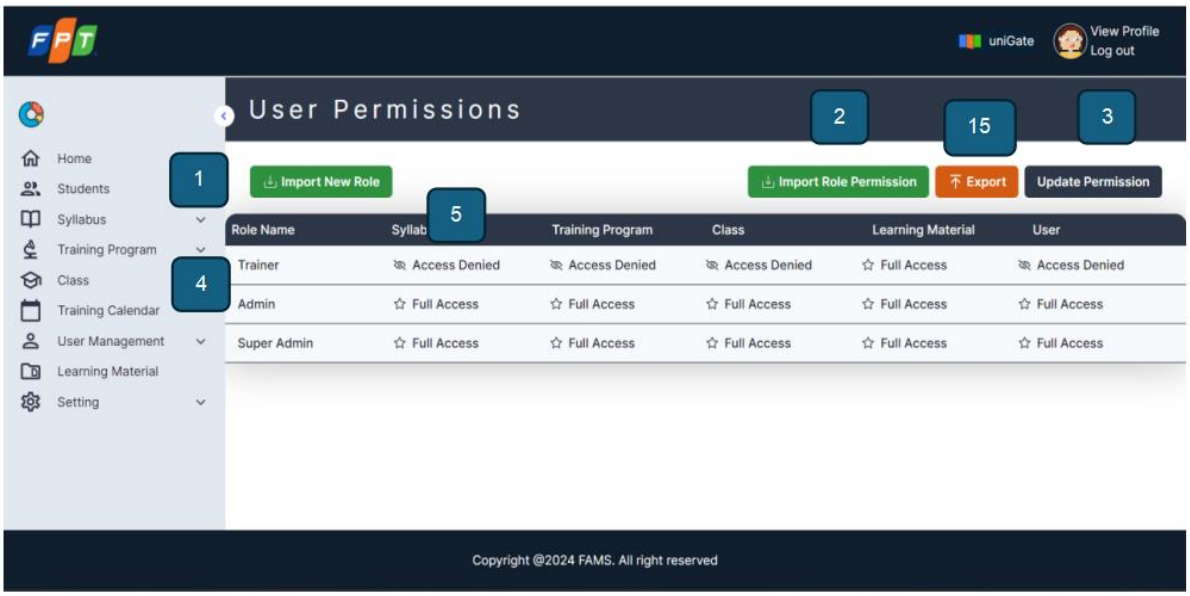


Figure 214 - User's Permission Page UI 1

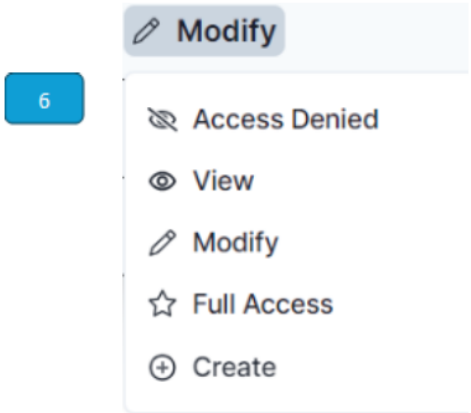


Figure 215 - Permission Menu - Page UI 2

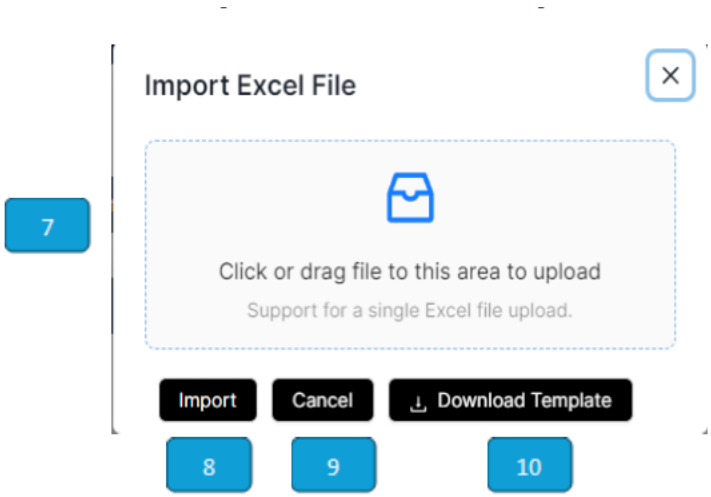


Figure 216 - Import New Role Pop Up - User's Permission Page UI 3

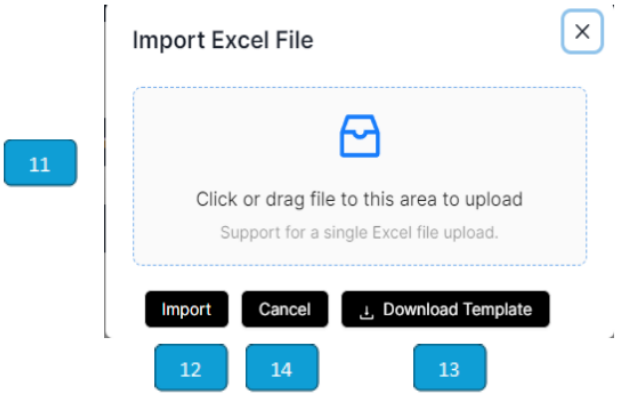


Figure 217 - Import Update Permission Pop Up - User's Permission Page UI 4

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
4	User's Permission list	Show User's Permission list	No	Yes	Table	Table data
6	Choose Permission	Choose a permission for this role	no	Yes	Dropdown	Text
7	Upload area	Area to upload excel file to import	No	Yes	Drag and drop area	Excel file

11	Upload area	Area to upload excel file to import	No	Yes	Drag and drop area	Excel file

Table 18 - <User Interface-Field> User's Permission Page

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Import New Role	Click to open Import New Role Pop up	N/A	Go to Import New Role Pop up
2	Import Update Permission	Click to open Import Update Permission Pop up	N/A	Go to open Import Update Permission Pop up
3	Update Permission	Click to update permission	N/A	Go to Add User Pop Up
5	Modify User's permission	Click to open permission menu	N/A	Sent request to system to update permission. Show message when success import
8	Import	Click to Import User from excel file	N/A	Sent request to system to Import a New Role Show message when success import Show all wrong verify when fails import
9	Cancel	Click to cancel Import User	N/A	Close pop up Go to User Management Page
10	Download template	Click to download template	N/A	Download ImportRoleTemplate.xlsx
12	Import	Click to Import User from excel file	N/A	Sent request to system to Import Update User's Permission Show message when success import Show all wrong verify when fails import
13	Cancel	Click to cancel Import User	N/A	Close pop up Go to User Management Page

14	Download template	Click to download template	N/A	Download ImportUser'sPermissionTemplate.xlsx
15	Export	Click to export Role Permission List	N/A	Download ExportUserPermissionList.xlsx

Table 19 - <User Interface-Button/Hyperlinks> User's Permission Page

5.1.5. User Calendar

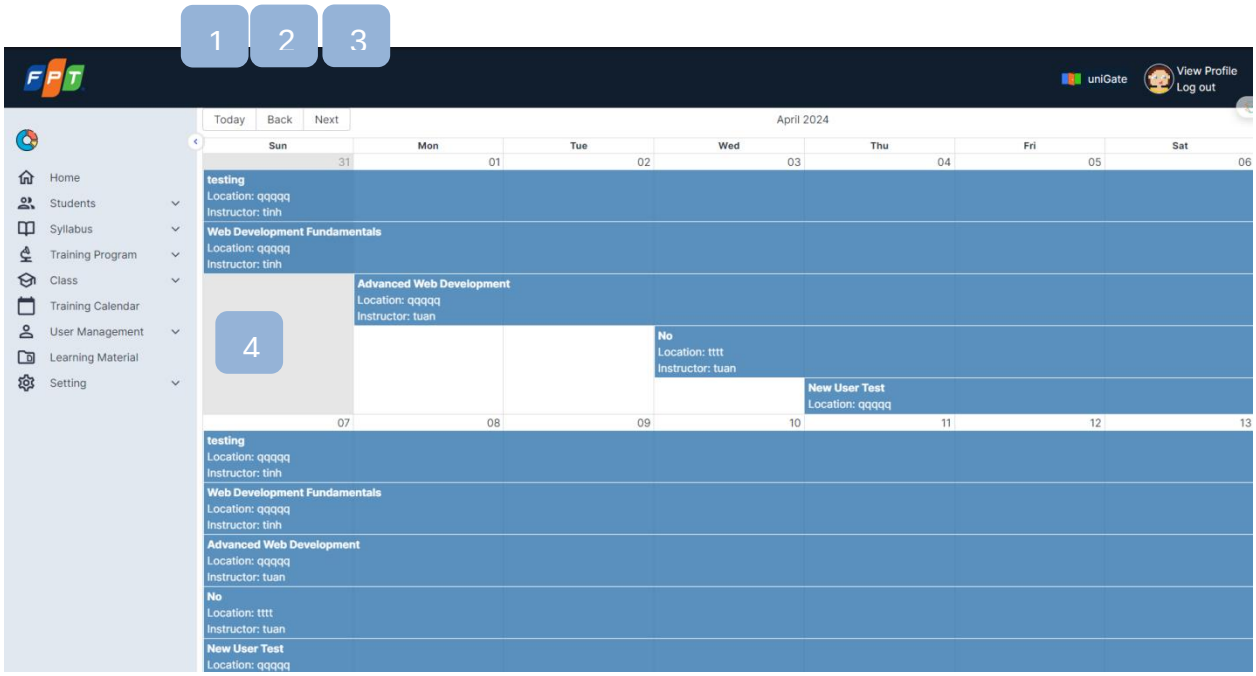


Figure 218 - User Calendar

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
4	User Calendar	Show the calendar of current user	Yes	Yes	Big Calendar	Event

Table 20 - <User Calendar-Field> User Calendar

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Today	Move to the current day	N/A	Show the month contain the
2	Back	Move to the previous month	N/A	Navigate to user file explorer

3	Next	Move to the next month	N/A	Navigate to training program list

Table 21 - <User Calendar-Button/Hyperlinks> User Calendar

5.2. Class Management Interface

5.2.1. Create Class

Figure 219 - Create Class

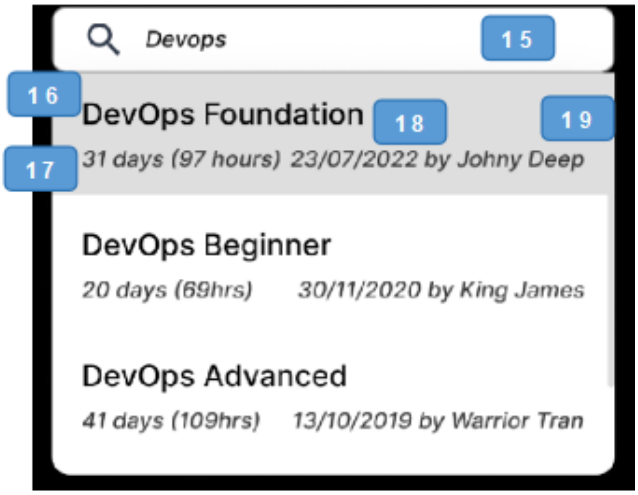


Figure 220 - Training program search field - Create Class

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Training program search field	User enter training program name for searching training program	No	Yes	TextBox	String	N/A
2	Mật Khẩu	Password to login	No	Yes	TextBox	String	N/A
5	Field training program	Display the field of trainer	No	Yes	TextBox	String	N/A
6	Field attendee list	Display the field of trainer	No	Yes	TextBox	String	N/A
9	General field	Dropdown info for user to input	No	Yes	Single choice dropdown	String & Number	N/A
10	Time frame	Dropdown info for user to input	No	Yes	Single choice dropdown	Number	N/A

11	Attendance	Single choice dropdown	No	Yes	Single choice dropdown	Number & String	N/A
13	Syllabus status	Display syllabus status	Yes	No	Label	String	N/A
14	Show time of class	Display the time of class	Yes	No	Label	String	N/A
15	Search recommend list	Show list training program searched by input keyword	No	Yes	Textbox	String	N/A
16	Training program name	Display Training program name	Yes	No	Label	String	N/A
17	Training program duration	Training program total duration. Display follow format: [total days] day ([total hours] hours)	Yes	No	Label	Number	N/A
18	Training program last modified	Training program last modified information. Whenever there is any change to this screen it get updated Follow format: Modified on [date] by [account name]	Yes	No	Label	String	N/A

Table 22 - <User Interface-Field> Create Class

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome

2	Cancel button	Discard all data and transit to Class list page	N/A	Navigate to class list page
3	Save as draft	Go to save the input	N/A	Navigate to class list page
4	Next button	Go to edit syllabus of training program for class	N/A	Navigate to Register page
5	Field training program	Display the field of trainer	N/A	N/A
7	Budget	N/A	N/A	N/A
8	Other	N/A	N/A	N/A
12	More Option	N/A	N/A	N/A

Table 23 - <User Interface-Button/Hyperlinks> Create Class

5.2.2. Selected Training Program and Show Syllabus

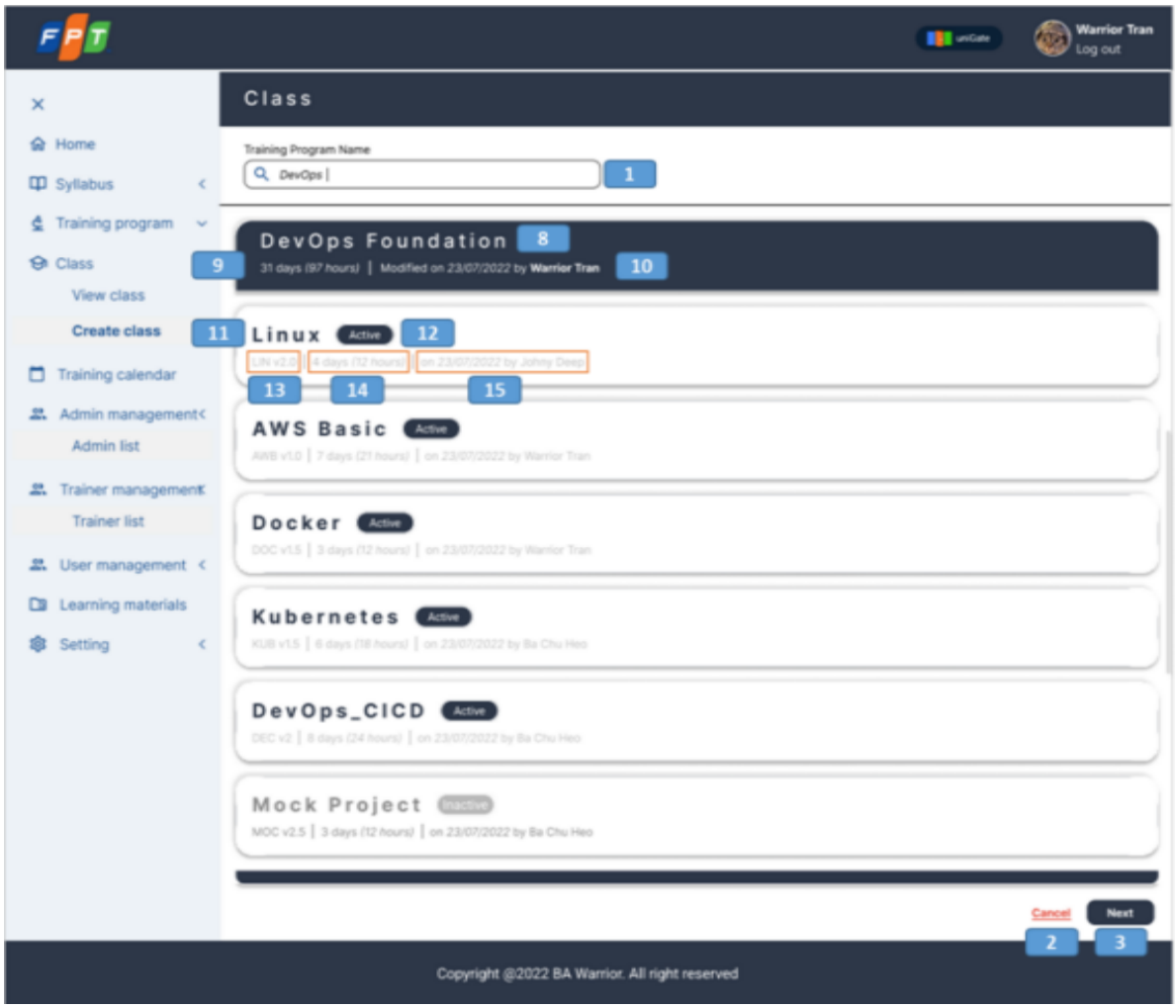


Figure 221 - Selected Training Program and Show Syllabus

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
8	Training program name	Display Training program name	Yes	Yes	Label	String	N/A

9	Training program duration	Display Training program duration	Yes	Yes	Label	Number & String	N/A
10	Training program last modified	Display Training program last modified	Yes	Yes	Label	String	N/A
11	Syllabus name	Display syllabus name Transit to syllabus detail page	Yes	Yes	Hyperlink	String	N/A
12	Syllabus status	Display syllabus status	Yes	Yes	Label	String	N/A
13	Syllabus short name and version	Display short name and version of syllabus follow format: [short name] [version]	Yes	Yes	Label	String and number	N/A
14	Syllabus duration	Display syllabus duration, follow format: [total days] day ([total hours] hours)	Yes	Yes	Label	String and Number	N/A
15	Syllabus modification history	Display last update information, follow format: on [date] by [account name]	Yes	Yes	Label	String	N/A

Table 24 - <User Interface-Field> Selected Training Program and Show Syllabus

5.2.3. Edit Training Detail

Figure 222 - Edit Training Detail

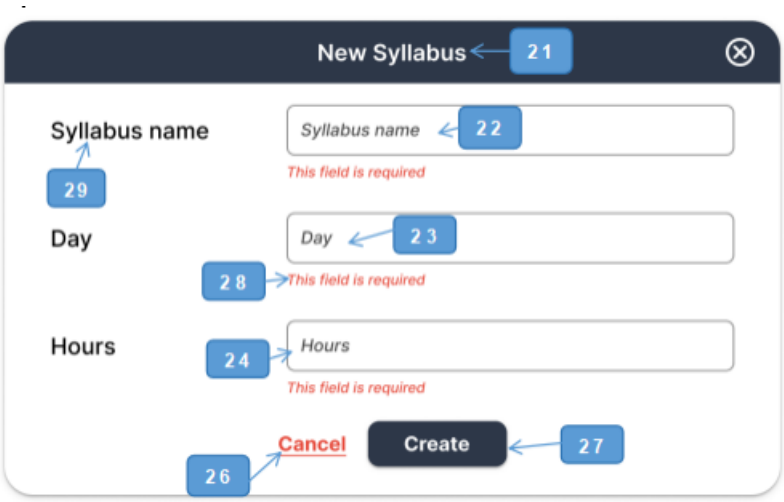


Figure 223 - Pop-up add new syllabus - Edit Training Detail

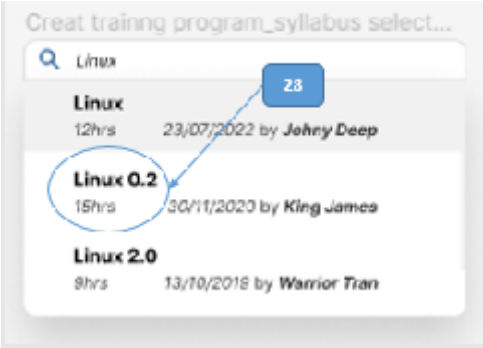


Figure 224 - Select syllabus - Edit Training Detail

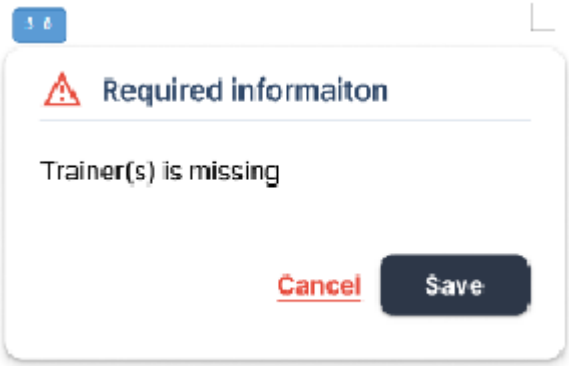


Figure 225 - Training program_syllabus - Edit Training Detail

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Training Class Name	Training Class Name	Yes	Yes	Label	String	N/A
2	Training Program Name	Training Program Name	Yes	Yes	Label	String	N/A
3	Training program total days	Training program total date	No	Yes	Label	String	N/A
4	Training program total duration	Training program total duration	No	Tes	Label	String	N/A
5	Training program last modified date	Training program last modified date	No	Yes	Label	String	N/A
6	Training program last modified by	Training program last modified by	No	Yes	Label	String	N/A
7	Syllabus Name	Syllabus Name	Yes	Yes	Label	String	N/A
8	Syllabus Status	Syllabus Status	Yes	Yes	Label	String	N/A
9	Syllabus code	Syllabus code	Yes	Yes	Label	String	N/A
10	Syllabus version	Syllabus version	Yes	Yes	Label	String	N/A
11	Syllabus total days	Syllabus total days	Yes	Yes	Label	String	N/A
12	Syllabus total duration	Display the duration of a Syllabus.	Yes	Yes	Label	String	N/A

13	Last Edit Date	Display the latest edit date of a Syllabus	Yes	Yes	Label	String	N/A
14	Editor	Display the latest editor's name	Yes	Yes	Label	String	N/A
15	Select	User can input to search available syllabus	Yes	Yes	Textbox	String	N/A
22	Syllabus Name	Syllabus Name	Yes	Yes	Textbox	String	N/A
23	Day	Activity days	Yes	Yes	Textbox	String	N/A
24	Hours	Activity Hours	Yes	Yes	TextBox	String	N/A
25	Name Field	Display Field to write	Yes	Yes	Label	String	N/A
28	Error Message	Show the require field for user	N/A	N/A	Label	String	N/A
29	Training program_syllabus	Training program_syllabus retrieved from keyword searching on select textbox	N/A	N/A	Label	String	N/A

Table 25 - <User Interface-Field> Edit Training Detail

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
16	Back	Previous step with saving new changes	N/A	N/A
17	Cancel	Previous step without saving.	N/A	N/A

18	Save	Redirect to "Add training program details for each day" page	N/A	N/A
19	Delete Syllabus	Pop up confirm dialog to confirm delete Syllabus	N/A	N/A
20	Add Syllabus	Display pop-up add new syllabus	N/A	N/A
21	New Syllabus	Pop-up's name	N/A	N/A
26	Cancel	Previous step without saving.	N/A	N/A
27	Save	Saving info of new Syllabus and back to the training program page of class management	N/A	N/A
30	Save as draft button	Save Trainer as Trainer class. Accept invalid value.	N/A	N/A

Table 26 - <User Interface-Button/Hyperlinks> Edit Training Detail

5.2.4. Class List Page

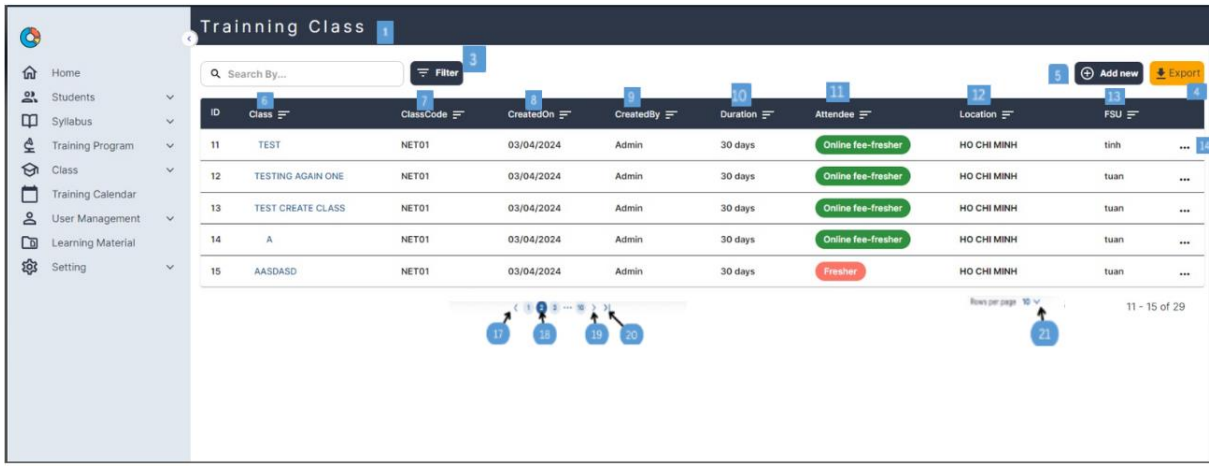


Figure 226 - Class List Page

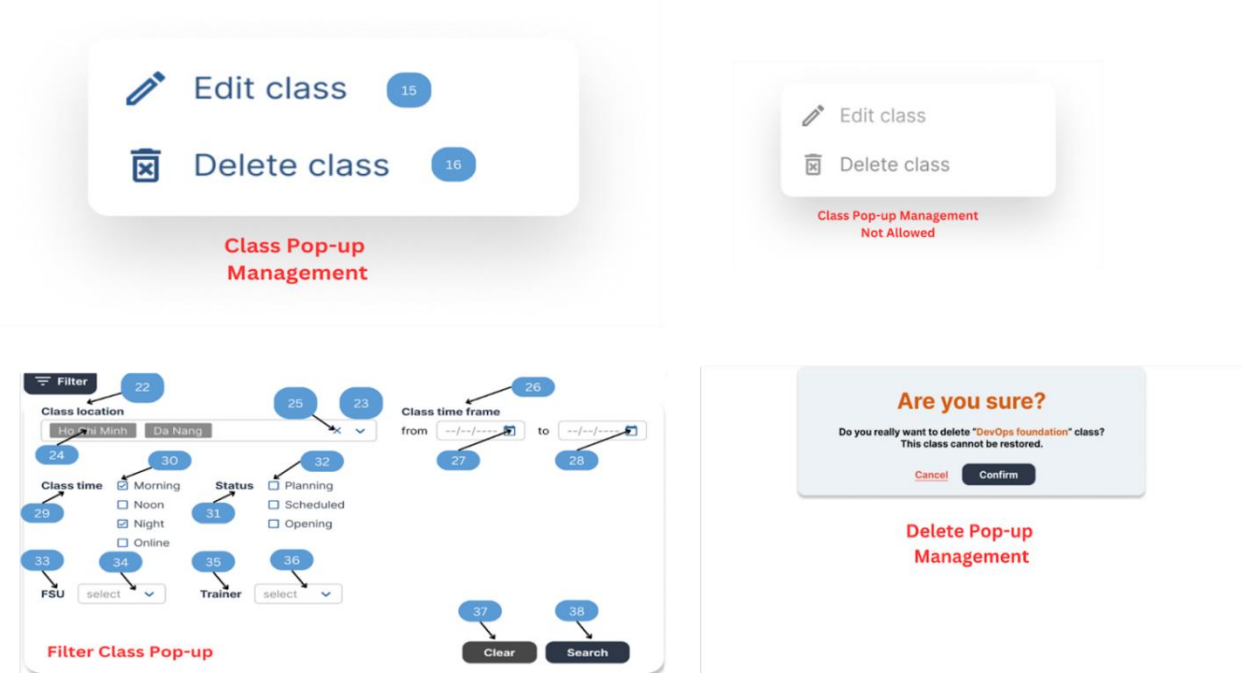


Figure 227 - Class Pop-up Management and Filter Class Pop-up - Class List Page

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Training Class	Training Class	Yes	Yes	Label	String	N/A
2	Search box	Enter keywords to search	No	Yes	TextBox	String	N/A
3	Filter	Display Class Filter Pop-up	No	Yes	TextBox	String	N/A
4	Export	Export Class Management info show to file	No	Yes	Label	String	N/A
6	Class	Display Training Class Name	Yes	Yes	Label	String	N/A

7	Class Code	Display Training Class Code	Yes	Yes	Label	String	N/A
8	Create on	Display Training Class Created on	Yes	Yes	Datetime picker	Format: dd/MM/yyyy	N/A
9	Created by	Display Training Class Created by	Yes	Yes	Label	String	N/A
10	Duration	Display Training Class Created by	Yes	Yes	Label	String	N/A
11	Status	Display Status of Class (Planning, Opening, Deactive, and Finished)	Yes	Yes	Label	String	N/A
12	Location	Display Training Class Location	Yes	Yes	Label	String	N/A
13	FSU	Display Training Class FSU	Yes	Yes	Label	String	N/A
22	Class location	Class location name	Yes	Yes	Label	String	N/A
23	Class location dropdown list	This drop down list allows user to select location to	No	Yes	Multiple choice dropdown		N/A

		search class.					
24	Location name keyword	Display name of location	Yes	Yes	Label	String	N/A
26	Class Time Frame	Class Time Frame to pick start day and end date	Yes	Yes	Label	String	N/A
27	Class Time From	This field allows user to input actual start date to search class.	No	Yes	Date Time	Date format: DD-MM-YYYY.	N/A
28	Class Time To	This field allows user to input actual start date to search class.	No	Yes	Date Time	Date format: DD-MM-YYYY.	N/A
29	Class Time	Class time label	Yes	Yes	Label	String	N/A
30	Class Time Selection	Class time includes Morning, Noon, Night, Online	No	Yes	Checkbox	Checkbox	N/A
31	Status	Status label	Yes	Yes	Label	String	N/A
32	Status selection	Status includes Planning, Openning, Closed	No	Yes	Checkbox	Checkbox	N/A
33	FSU	Fsoft Stratetry	Yes	Yes	Label	String	N/A

		Units					
35	Trainer	Trainer selection	Yes	Yes	Label	String	N/A

Table 27 - <User Interface-Field> Class List Page

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
5	Add Class	Click to move to the Create Class Page	N/A	Move to /Create Class page
14	Class Popup Management	Class pop-up management will be displayed	N/A	Pop up a box with 2 options to choose
15	Edit Class	Click to redirect to update the chosen training class page	N/A	Navigate to Edit Class page
16	Delete class	Modal confirmation to delete class shown up	N/A	Delete the chosen class
17	<	Click to move to the prev page.	N/A	Navigate to previous page
18	Page Number	Click to change to a specific number of page.	N/A	Navigate to the specific page
19	>	Click to move to the next page.	N/A	Navigate to next page
20	>||	Click to move to the last page.	N/A	Navigate to the last page
21	Rows per page	Choose the number of records displayed in 1 page.	N/A	Dropdown a list of number to choose how many numbers of records will be

				displayed in 1 page
25	Button "x" besides location name keyword	Removed location name keyword	N/A	Remove the location that have chosen
34	FSU select dropdown list	The place to choose FSU of class	N/A	Drop down a list to choose FSU
36	Trainer select dropdown list	Single choice dropdown	N/A	Drop down a list to choose Trainer
37	Clear	Clear all pick-up manual before, set all field to be original default	N/A	Clear all the choices in the Filter Popup
38	Search	Close the modal and the record data which is appropriate with field selection in modal box will be displayed on screen Class List	N/A	Send request to give back a filtered list

Table 28 - <User Interface-Button/Hyperlinks> Class List Page

5.2.5. Class Detail

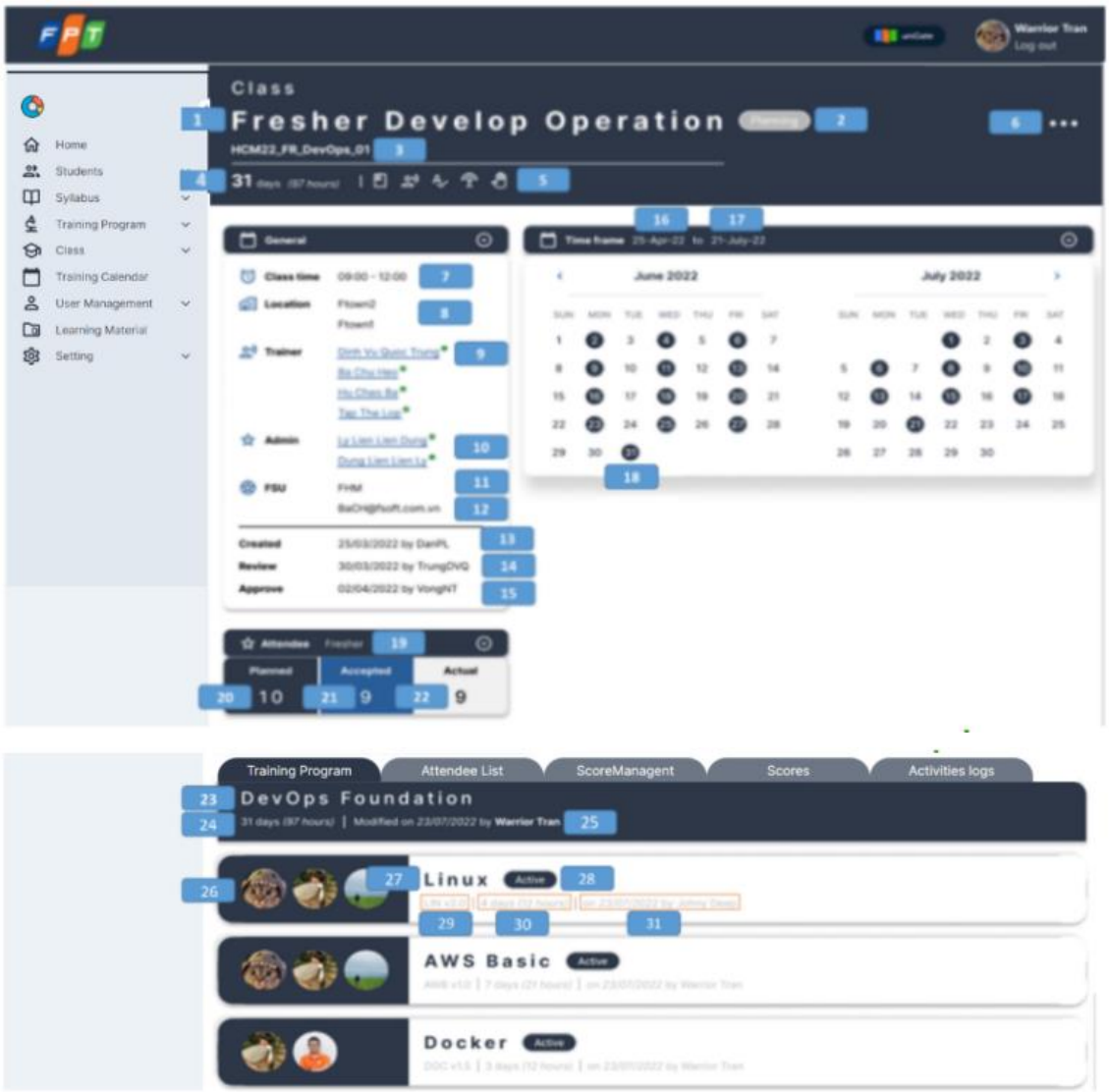


Figure 228 - Class Detail

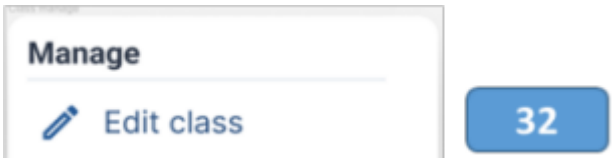


Figure 229 - Edit class - Class Detail

Figure 230 - Trainer/Admin phone number and email - Class Detail

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Class name	Display class name	Yes	No	Label	String	N/A
2	Class status	Display class status Value list: Planning, Opening, Finished, Deactive	Yes	No	Label	String	N/A
3	Class code	Class code	Yes	No	Label	String	N/A
4	Class duration	Display follow format: [total days] day ([total hours] hours)	Yes	No	Label	Int	N/A
5	Delivery type	Display delivery type symbol Show delivery type name on hover	Yes	No	Image	String	N/A
7	Class time	Show start time and end time of class in day,	Yes	No	Label	Time Only	N/A

		follow format: [start time] - [end time]					
8	Location	Display location of class	Yes	No	Label	String	N/A
11	FSU name	FSU's name organizing a class	Yes	No	Label	String	N/A
12	FSU contact point	FSU contact information	Yes	No	Label	String	N/A
13	Created	Display creation information, follow format: [date] by [account name]	Yes	No	Label	String and date time	N/A
14	Review	Display review information, follow format: [date] by [account name]	Yes	No	Label	String and date time	N/A
15	Approved	Display approve information, follow format: [date] by [account name]	Yes	No	Label	String and date time	N/A
16	Start date	Display class start date, format: dd/MM/yyyy	Yes	No	Label	Date only	N/A
17	End date	Display class end date,	Yes	No	Label	Date only	N/A

		format: dd/MM/yyyy					
18	Highlighted learning day	High light learning day	Yes	No	Label	String	N/A
19	Attendee type	Display attendee type	Yes	No	Label	String	N/A
20	Planned attendee number	Display the number of planned students joining a class	Yes	No	Label	int	N/A
21	Accepted attendee number	Display the number of accepted students for a class	Yes	No	Label	int	N/A
22	Actual attendee number	Display the number of actual students joining a class	Yes	No	Label	int	N/A
23	Training program name	Display training program name	Yes	No	Label	String	N/A
24	Training program duration	Display training program duration, follow format: [total days] day ([total hours] hours)	Yes	No	Label	Int	N/A
25	Training program modification	Display last update information, follow format: Modified on [date] by	Yes	No	Label	String and date time	N/A

		[account name]					
26	Trainer avatar	Display trainer's avatar of syllabus	Yes	No	Image	String	N/A
27	Syllabus name	Displays the syllabus name and a link to the syllabus details	Yes	No	Image	String	N/A
28	Syllabus status	Display syllabus status	Yes	No	Label	String	N/A
29	Syllabus short name and version	Display version of syllabus	Yes	No	Label	String	N/A
30	Syllabus duration	Display syllabus duration, follow format: [total days] day ([total hours] hours)	Yes	No	Label	Int	N/A
31	Syllabus modification history	Display last update information, follow format: on [date] by [account name]	Yes	No	Label	String and date time	N/A
33	Trainer/Admin phone number	Display trainee/admin phone number	Yes	No	Label	String	N/A
34	Trainer/Admin email	Display trainee/admin email	Yes	No	Label	String	N/A

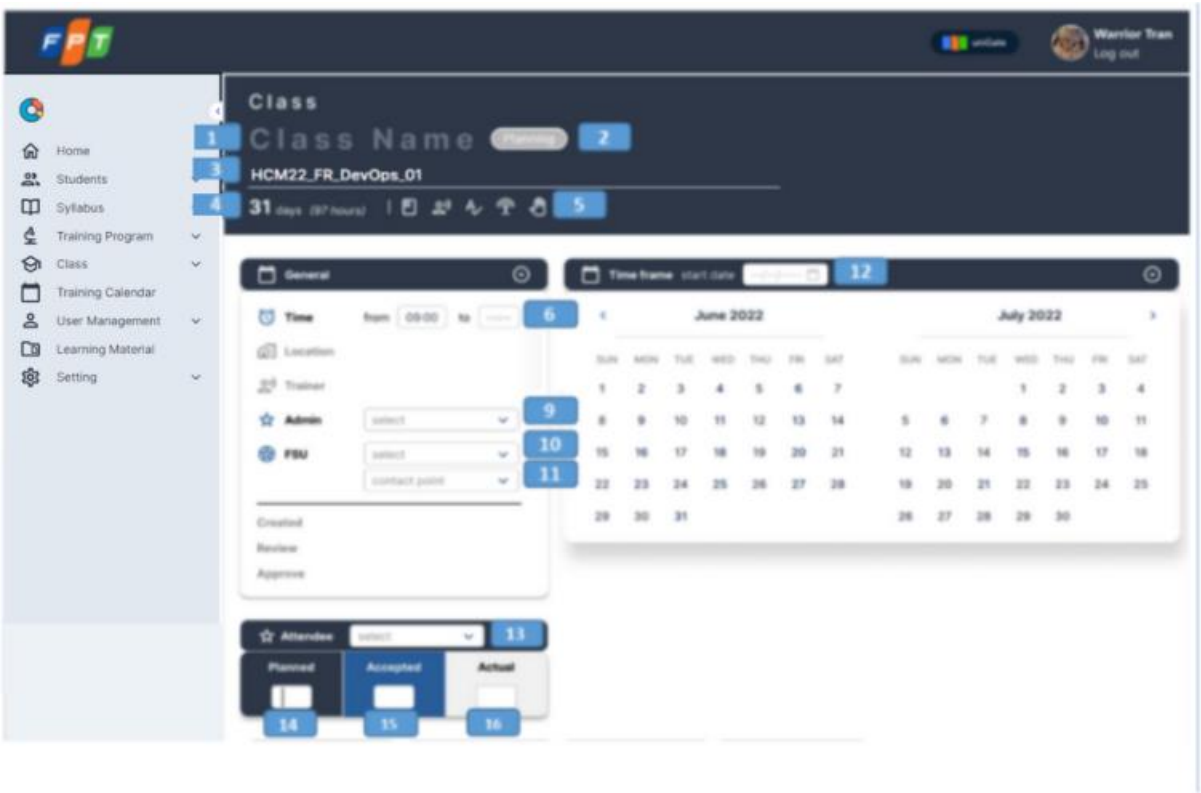
Table 29 - <User Interface-Field> Class Detail

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
6	Manage menu button	Show manage menu	N/A	Edit class
9	Trainer name	Path of trainer profile	N/A	Transit to trainer profile page
10	Admin name	Path of admin profile	N/A	Transit to admin profile page
32	Edit class	Path of admin profile	N/A	Transit to edit class page

Table 30 - <User Interface-Button/Hyperlinks> Class Detail

5.2.6. Edit Class



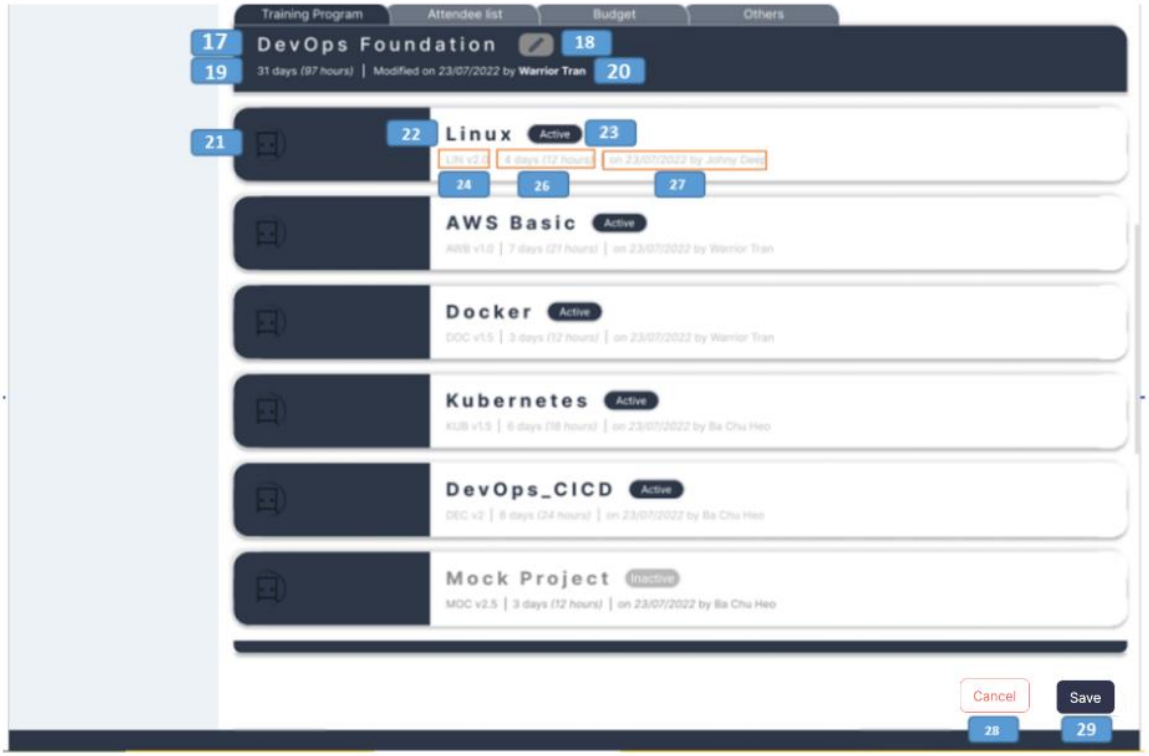


Figure 231 - Edit Class

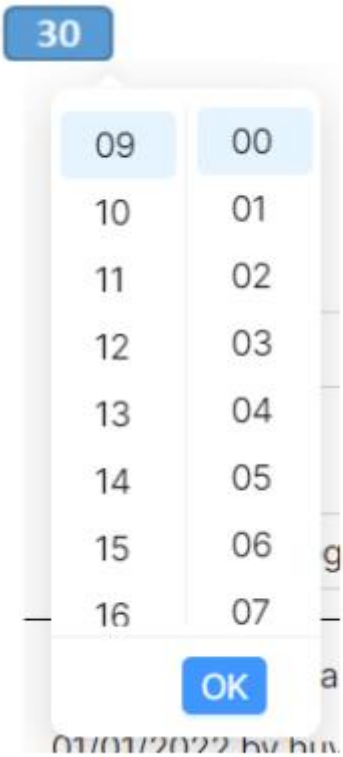


Figure 232 - Time picker drop-down - Edit Class

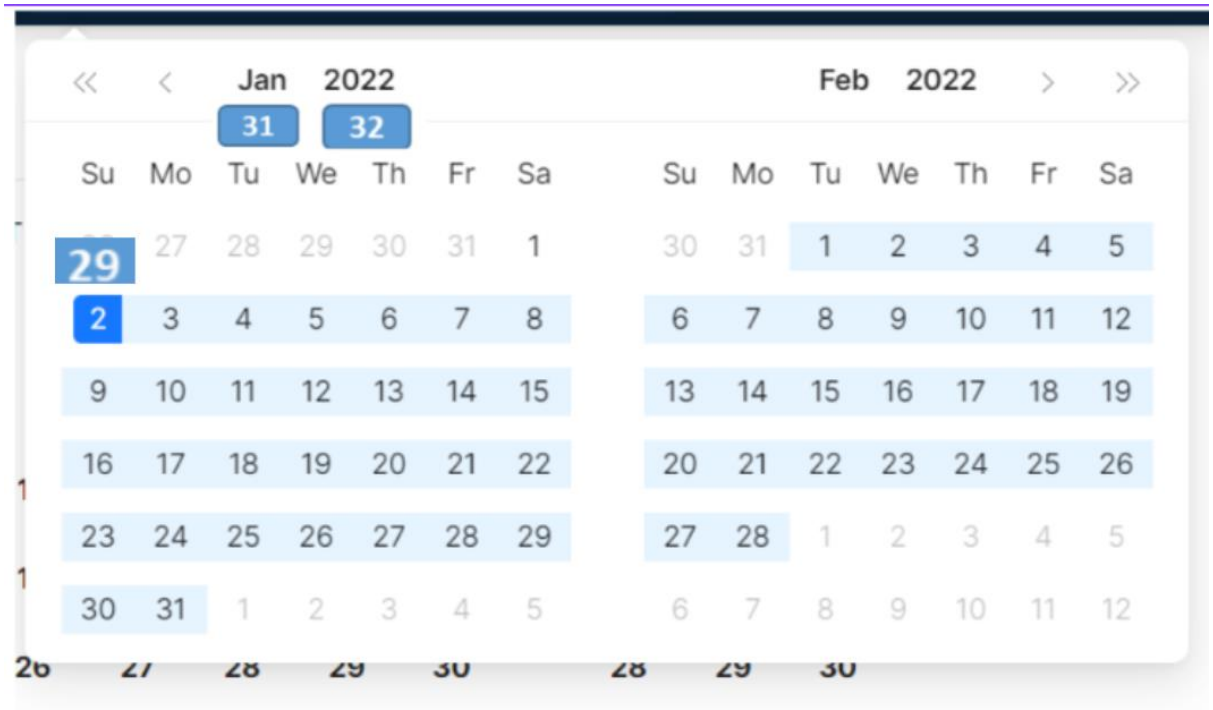


Figure 233 - Pop-up month,year,date picker - Edit Class

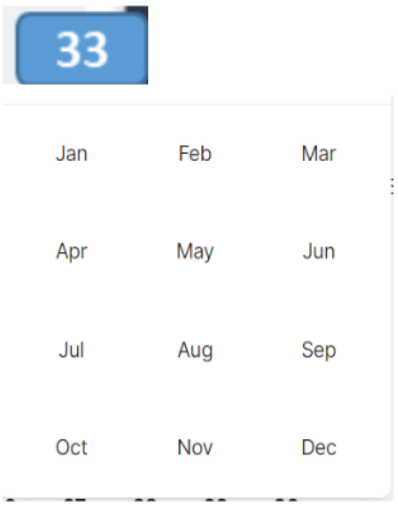


Figure 234 - Month drop-down list - Edit Class

Figure 235 - Year drop-down list - Edit Class

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Class name	Display class name	Yes	No	Label	String	N/A
2	Class status	Display class status Value list: Planning,Deactive, Finished,Opening Default value: Planning	Yes	No	Label	String	N/A
3	Class code	Class code	Yes	No	Label	String	N/A
4	Class duration	Display follow format: [total days] day ([total hours] hours)	Yes	No	Label	Int	N/A

5	Delivery type	Display delivery type symbol Show delivery type name on hover	Yes	No	Image	String	N/A
6	Class time	Pick start time and end time of a learning day	No	Yes	Datetime picker	Time Only	N/A
9	Admin name	Allows users to select class admin(s) for class.	No	Yes	Multiple choice dropdown	String	N/A
10, 11	FSU name and contact point	FSU's name organising a class and contact information	No	Yes	Single choice dropdown	String	N/A
12	Start date	Allows users to input the start date of class. Format: dd/MM/yyyy	No	Yes	Datetime picker	Date only	N/A
13	Attendee type	Allow the user to select the attendee type. Value list: Intern,Fresher,Online fee-fresher,Offline fee-fresher	No	Yes	Single choice dropdown	String	N/A
14	Planned attendee number	Allow user enter planned attendee number	No	Yes	Textbox	Int	N/A

15	Accepted attendee number	Allow user enter accepted attendee number	No	Yes	Textbox	Int	N/A
16	Actual attendee number	Allow user enter actual attendee number	No	Yes	Textbox	Int	N/A
17	Training program name	Display training program name	Yes	No	Label	String	N/A
19	Training program duration	Display training program duration, follow format: [total days] day ([total hours] hours)	No	Yes	Label	Int	N/A
20	Training program modification history	Display last update information, follow format: Modified on [date] by [account name]	Yes	No	Label	String	N/A
21	Trainer avatar	Display trainer's avatar of syllabus	Yes	No	Image	String	N/A
22	Syllabus name	Display syllabus name	Yes	No	Label	String	N/A
23	Syllabus status	Display syllabus status	Yes	No	Label	String	N/A

24	Syllabus version	Display version	Yes	No	Label	String	N/A
26	Syllabus duration	Display syllabus duration, follow format: [total days] day ([total hours] hours)	Yes	No	Label	Int	N/A
27	Syllabus modification history	Display last update information, follow format: on [date] by [account name]	Yes	No	Label	String and date time	N/A
29	Date picker	Allow user pick date value from calendar pop-up	No	Yes	Datetime picker	String	N/A
30	Time picker drop-down	Allow user pick time.	No	Yes	Datetime picker	String	N/A
33	Month drop-down list	Allow user pick month value	No	Yes	Single choice dropdown	String	N/A
34	Year drop-down list	Allow user pick year value	No	Yes	Single choice dropdown	String	N/A

Table 31 - <User Interface-Field> Edit Class

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome

18	Training program name edit button	Allow user edit training program name when click	N/A	Select this training program drop-down list
28	Cancel button	Discard all data	N/A	Discard all data
29	Save	Save class as draft class. Accept invalid value.	N/A	Save class as draft class. Accept invalid value.
31	Month picker	Display [Month drop-down list]	N/A	Display [Month drop-down list]
32	Year picker	Display [Year drop-down list]	N/A	Display [Year drop-down list]

Table 32 - <User Interface-Button/Hyperlinks> Edit Class

5.3. Syllabus Management Interface

5.3.1. Import syllabus

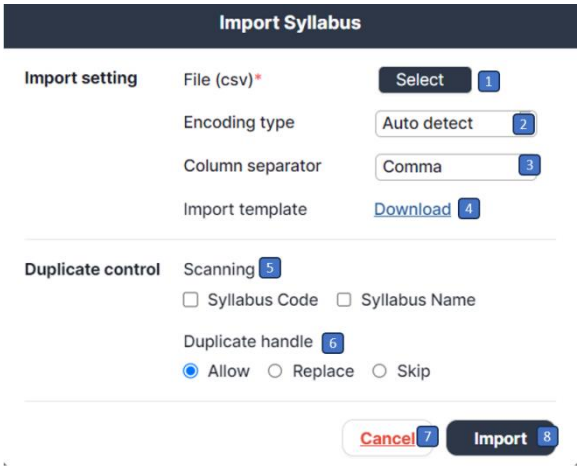


Figure 236 - Import syllabus UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
2	Encoding type	Encoding type of the file	No	Yes	Drop down list	String

3	Column separator	Column separator of the file	No	Yes	Drop down list	String
5	Scanning	Scanning the version and code of the file	No	Yes	Check box	String
6	Duplicate handle	Option to change when file duplicate	No	Yes	Radio check box	String

Table 33 - <User Interface-Field> Import syllabus

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Select	Select file to import	Only excel or csv file	Show file name has just been imported
4	Download	Download the syllabus template	N/A	Navigate to user file explorer
7	Cancel	Cancel import process	N/A	Navigate to syllabus list
8	Import	User import new data	N/A	Show success message

Table 34 - <User Interface-Button/Hyperlinks> Import syllabus

5.3.2. View list syllabus

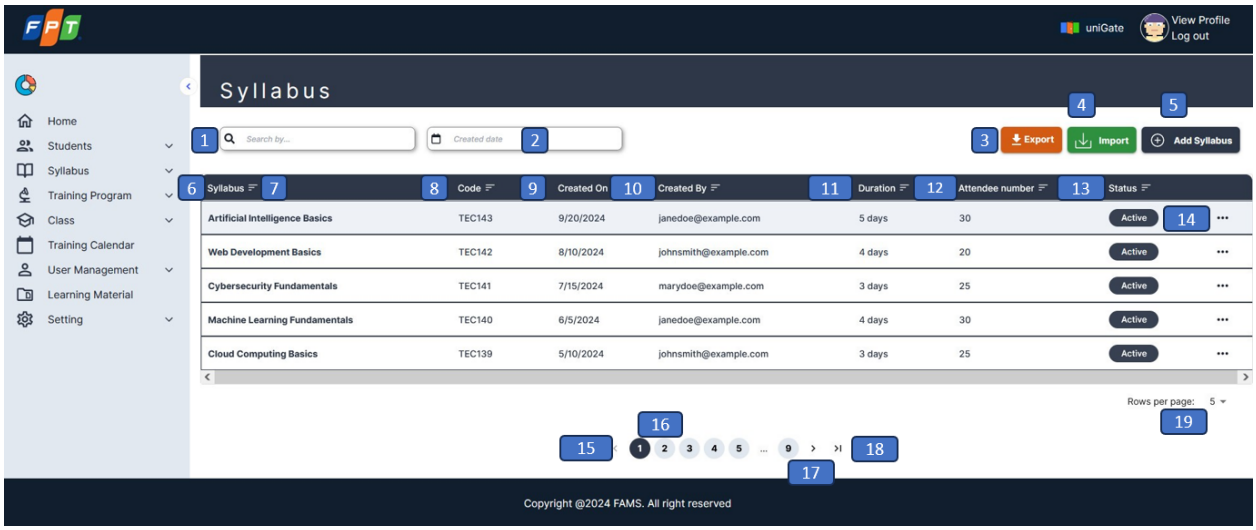


Figure 237 - View list syllabus UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Search by	User search syllabus detail	No	Yes	Text box	String
2	Created date	Search day created	No	Yes	Date time picker	Date time
6	Syllabus	Show syllabus list name	Yes	Yes	Label	String
8	Code	Show syllabus list code	Yes	Yes	Label	String
9	Created on	Show day created	Yes	Yes	Label	Date time
10	Created by	User name who creates syllabus	Yes	Yes	Label	String
11	Duration	Total duration of 1 syllabus	Yes	Yes	Label	Int
12	Attendee number	Show attendee number of syllabus	Yes	Yes	Label	Int
16	Status	Show status of syllabus	Yes	Yes	Label	String

Table 35 - <User Interface-Field> View list syllabus

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	Import	User use to import new data	N/A	Navigate to import syllabus modal
4	Export	User use to download syllabus list as csv file	N/A	Download notification
5	Add syllabus	Add new syllabus data	N/A	Navigate to create syllabus page
7	Sort	Sort list syllabus	N/A	Show list has been sorted
14	More	Open more functions to manage syllabus list	N/A	Navigate to syllabus menu

16	Page number	Click to change to a specific number of pages.	N/A	Change to others page
15	<	Click to move to the previous page.	N/A	Back 1 page.
17	>	Click to move to the next page.	N/A	Go to next page.
18	>||	Click to move to the last page.	N/A	Go to last page.
19	List per page	syllabus per page	N/A	Show number of syllabus list per page

Table 36 - <User Interface-Button/Hyperlinks> View list syllabus

5.3.3. Syllabus menu

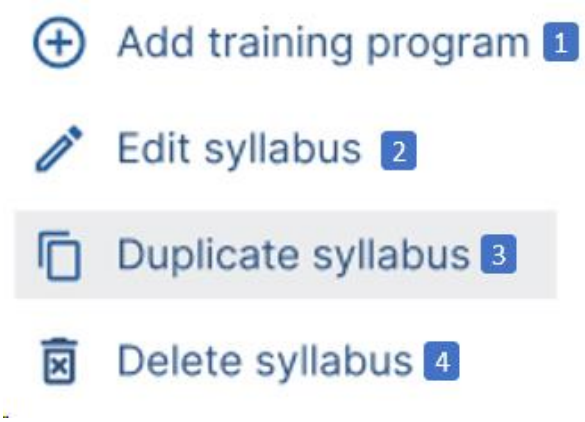


Figure 238 - Syllabus menu UI

Buttons/Hyperlinks:

No	Field Name	Description	Validation	Outcome
1	Add training program	Add more training program in syllabus	N/A	Navigate to create training program page
2	Edit syllabus	User update exist syllabus in list	N/A	Navigate to create syllabus page
3	Duplicate syllabus	Create a copy syllabus	N/A	Show success message.
4	Delete syllabus	Delete syllabus	Syllabus must be inactive	Show success message.

Table 37 - <User Interface-Button/Hyperlinks> Syllabus menu

5.3.4. Syllabus manage

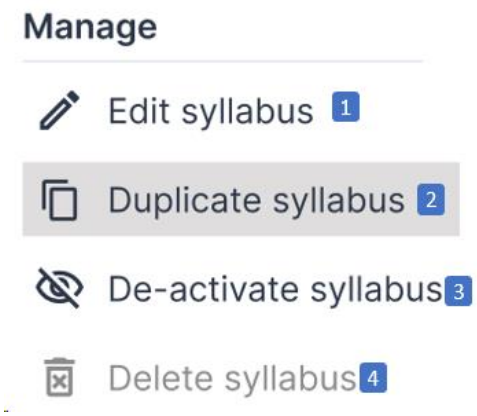


Figure 239 - Syllabus manage UI

Buttons/Hyperlinks:

No	Field Name	Description	Validation	Outcome
1	Edit syllabus	User update syllabus detail	N/A	Navigate to create training program page
2	Duplicate syllabus	Create a copy syllabus	N/A	Show success message.
3	De-activate syllabus	Change status of syllabus	N/A	Show success message.
4	Delete syllabus	Delete syllabus	Syllabus must be inactive	Show success message.

Table 38 - <User Interface-Button/Hyperlinks> Syllabus manage

5.3.5. Create Syllabus-General

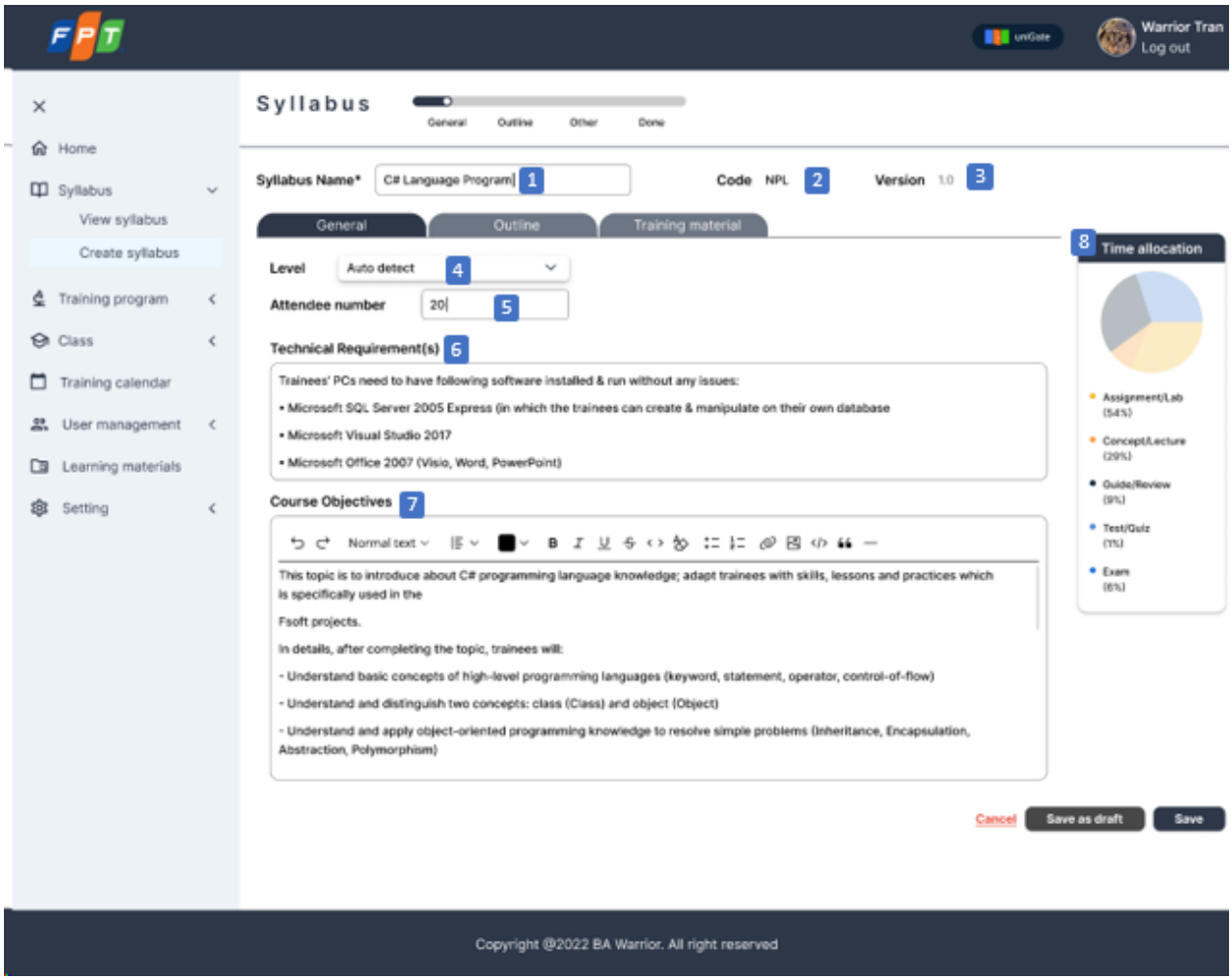


Figure 240 - Create Syllabus-General UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Syllabus Name	Input name of syllabus	No	Yes	Text box	String
2	Code	Syllabus code	Yes	Yes	Label	String
3	Version	Syllabus version	Yes	Yes	Label	String
4	Level	Display low to high level	No	Yes	Single choice dropdown	String
5	Attendee	Input number of	No	Yes	Text box	String

	number	attendees				
6	Technical Requirement (s)	Technical	Yes	Yes	Text area	String
7	Course Objectives	Total duration of 1 syllabus	Yes	Yes	Text area	String
8	Time allocation	Show the time percentage of each delivery type automatically calculated based on the Syllabus Outline data	Yes	Yes	Table	String

Table 39 - <User Interface-Field> Create Syllabus-General

5.3.6. Create Syllabus-Outline

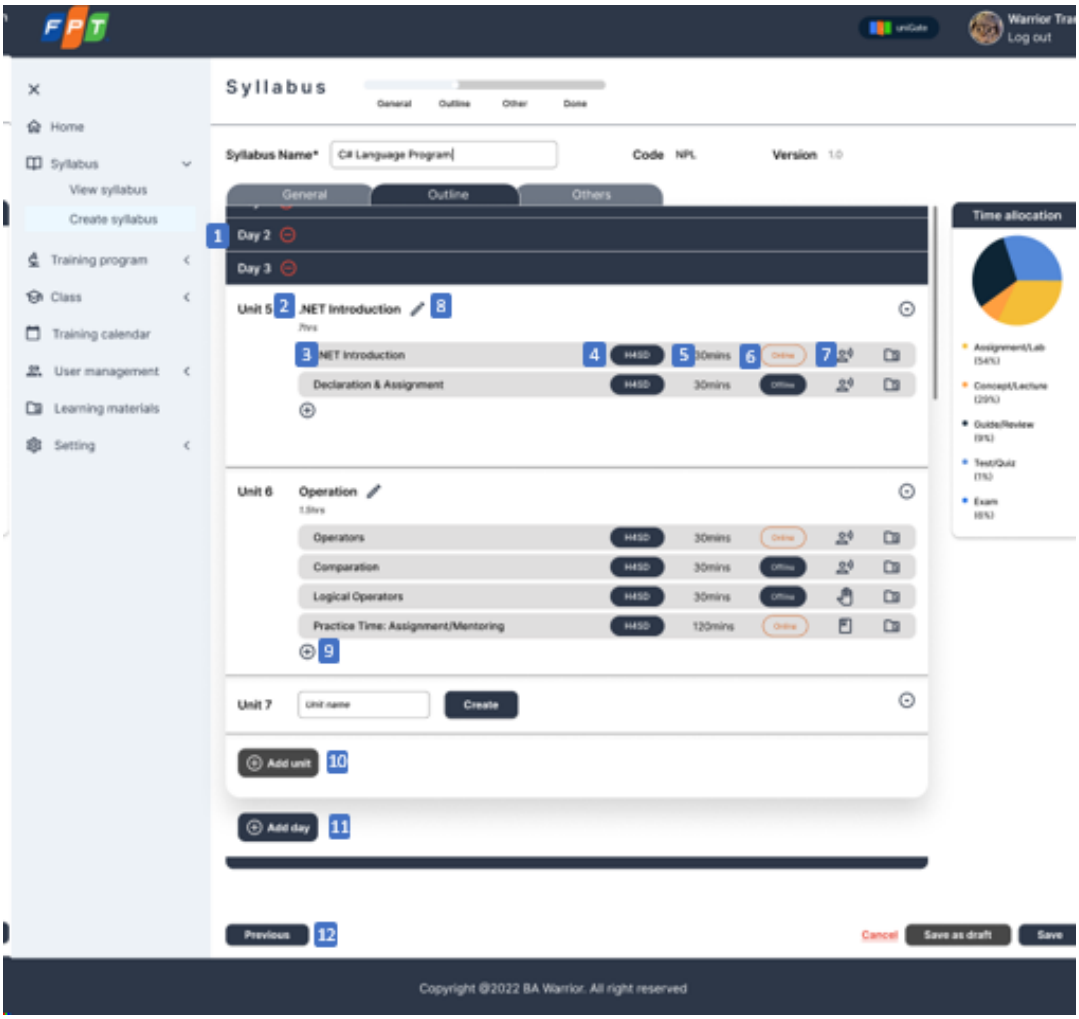


Figure 241 - Create Syllabus-Outline UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Day	Show auto-increment day number	Yes	Yes	Label	String
2	Unit name	Name of unit	No	Yes	Text box	String
3	Chapter name	Name of the chapter	No	Yes	Text box	String
4	Output Standard	What you will gain when you finished leaning this chapter	No	Yes	Single choice dropdown	String
5	Chapter duration	Time needed to finish learning this chapter	No	Yes	Text box	Int
6	Chapter type	Chapter type (online, offline). Default type: Online	No	Yes	Button	String
7	Delivery type	Delivery type of the chapter	No	Yes	Single choice dropdown	String

Table 40 - <User Interface-Field> Create Syllabus-Outline

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
8	Edit	Change name of the unit	N/A	Open text area to edit name
9	Add new content	Button to add 1 more input chapter template row to input new chapter	Required validate all field	Show success message
10	Add unit	Button to add 1 more unit template for user to input	N/A	One more unit is added below
11	Add day	Button to add 1 more day	N/A	One more day is added below
12	Previous	Go to the previous tab of the current tab	N/A	Navigate to general tab

Table 41 - <User Interface-Button/Hyperlinks> Create Syllabus-Outline

5.3.7. Create Syllabus-Edit Unit

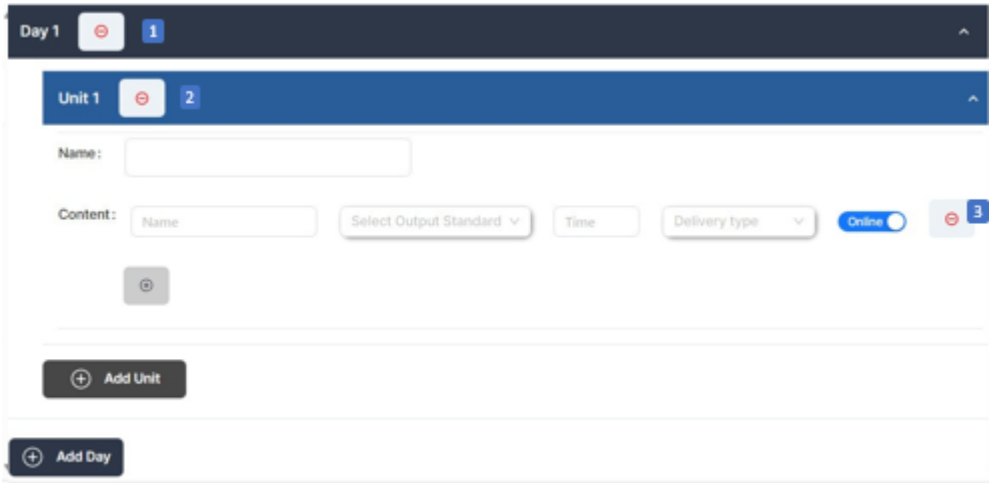


Figure 242 - Create Syllabus-Edit Unit UI

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Delete day	Delete the chosen day	N/A	Show success message
2	Delete unit	Delete the chosen unit and all its chapters that belong to	N/A	Show success message
3	Delete chapter	Delete all content of the chapter	N/A	Show success message

Table 42 - <User Interface-Button/Hyperlinks> Create Syllabus-Edit Unit

5.3.8. Create Syllabus-Other

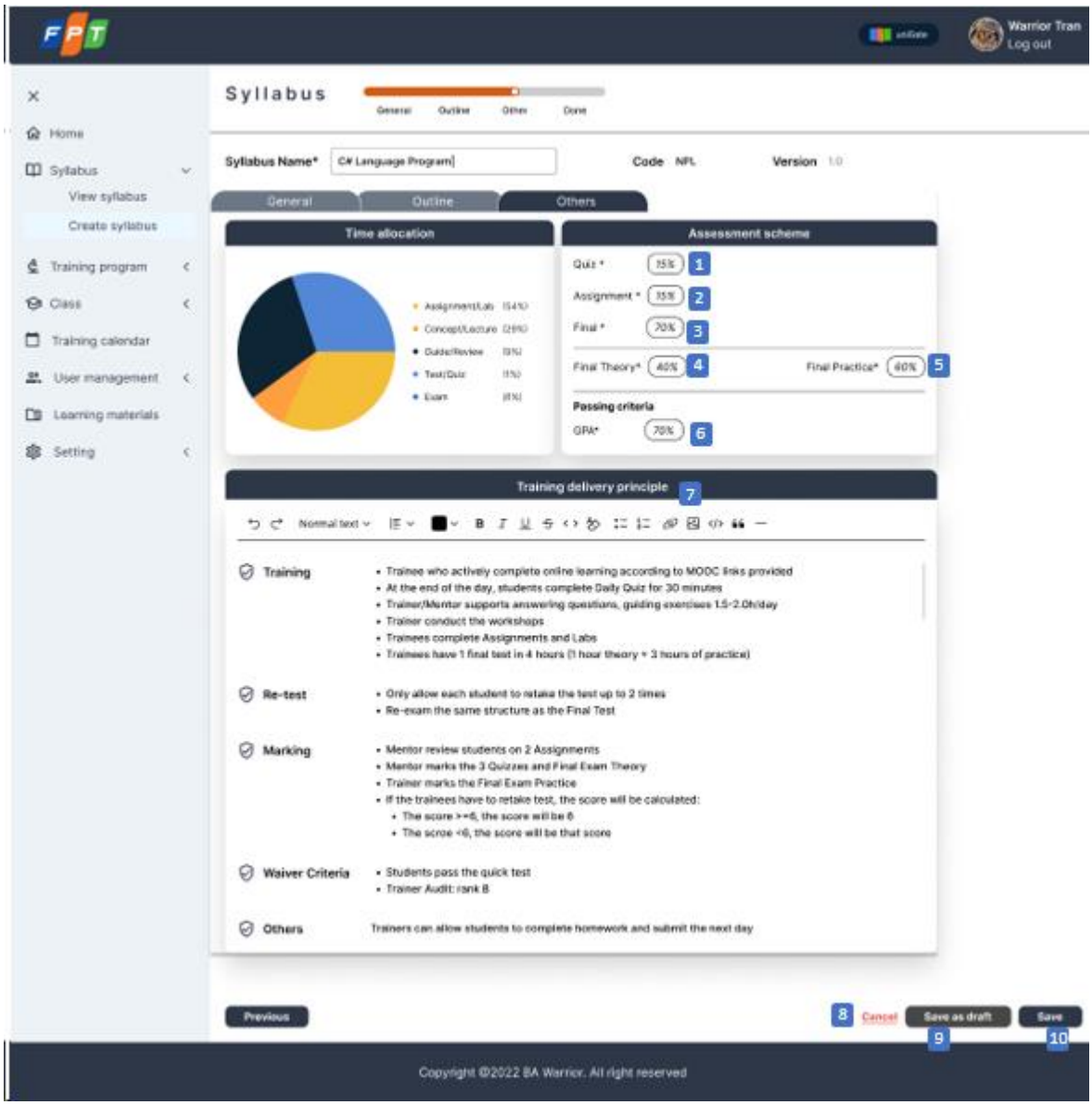


Figure 243 - Create Syllabus-Other UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Quiz	Percentage of final theory point that quiz point take	No	Yes	Text box	String
2	Assignment	Percentage of final theory point that assignment point take	No	Yes	Text box	String

3	Final	Percentage of final theory point that final point take	No	Yes	Text box	String
4	Final Theory	Percentage of total point that final theory point take.	No	Yes	Text box	String
5	Final Practice	Percentage of total point that final practice point take.	No	Yes	Text box	String
6	GPA	Percentage of total point need to pass this syllabus	No	Yes	Text box	String
7	Training delivery principle	Training delivery principle	No	Yes	Text area	String

Table 43 - <User Interface-Field> Create Syllabus-Other

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
8	Cancel	Cancel the creating syllabus process	N/A	Navigate to syllabus list page
9	Save as draft	Save the inputting syllabus as draft	N/A	Navigate to alert popup
10	Save	Create new syllabus	N/A	Show success message or alert popup.

Table 44 - <User Interface-Button/Hyperlinks> Create Syllabus-Other

5.3.9. Create Syllabus-Alert popup

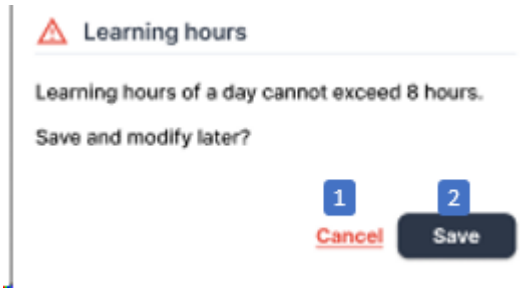


Figure 244 - Create Syllabus-Alert popup UI

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Cancel	Close the recommend save as draft popup	N/A	Navigate back to create process.
2	Save	Save the inputting syllabus as draft	N/A	Show success message.

Table 45 - <User Interface-Button/Hyperlinks> Create Syllabus-Alert popup

5.3.10. Training materials

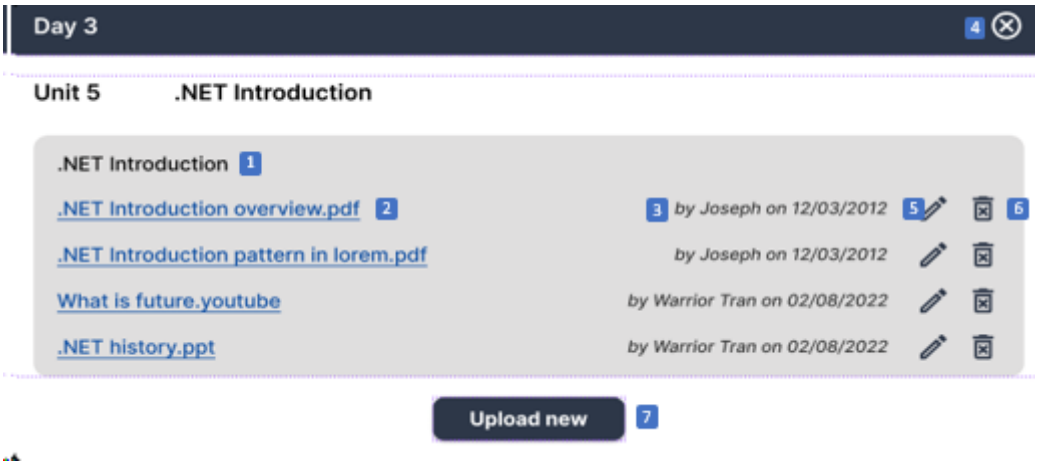


Figure 245 - Training materials UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Chapter name	Name of the chapter of those training material belong to	Yes	Yes	Label	String
2	Material name	Name of the file or link training material	No	Yes	Label	Hyper link
3	Modify information	Name of the file or link training material	Yes	Yes	Label	String

Table 46 - <User Interface-Field> Training materials

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome

4	Close	Close the training material popup	N/A	Close popup successfully.
5	Edit	Edit training materials	N/A	Navigate to user file explorer
6	Delete	Delete training material	N/A	Show success message.
7	Upload new	Upload new training material	N/A	Navigate to user file explorer.

Table 47 - <User Interface-Button/Hyperlinks> Training materials

5.4. Training Program Management Interface

5.4.1. Training Program List

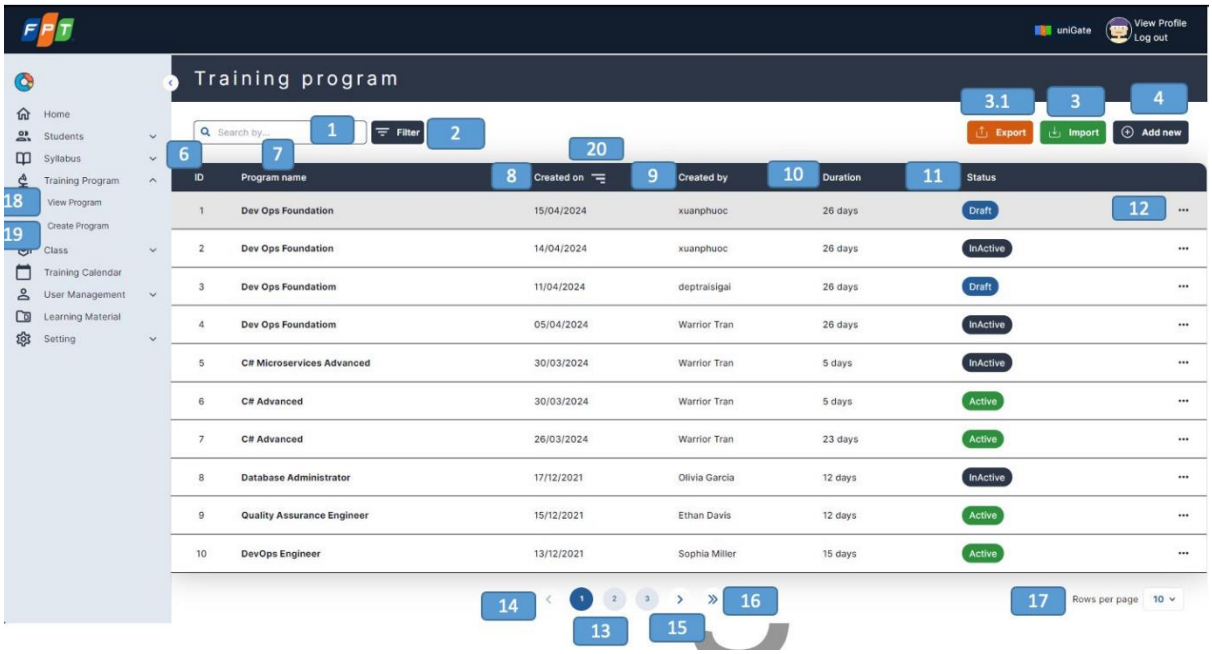


Figure 246 - Training Program List UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Search by	Name of the chapter of those training material belong to	No	Yes	Text box	String
6	ID	Name of the file or link training material	Yes	Yes	Label	String

7	Program name	Name of the file or link training material	Yes	Yes	Label	String
8	Created on	Display Training Program Created on	Yes	Yes	Label	Date time
9	Created By	Display Training Program Created by	Yes	Yes	Label	String
10	Duration	Display Training Program Duration	Yes	Yes	Label	String
11	Status	Display Training Program Status	Yes	Yes	Label	String

Table 48 - <User Interface-Field> Training Program List

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
2	Filter	Click to pop up filter table	N/A	Close popup successfully.
3	Import	Click to pop up import table	N/A	Navigate to user file explorer
3.1	Export	Click to download all record training program	N/A	Download file successfully
4	Add new	Click to redirect to create new training program page	N/A	Show success message.
20	Sort	Click to sort a specific field	N/A	Sort successfully
12	More	Click to pop up program management.	N/A	Popup program management table.
13	Page number	Click to change to a specific number of pages.	N/A	Change to others page
14	<	Click to move to the previous page.	N/A	Back 1 page.
15	>	Click to move to the next page.	N/A	Go to next page.

16	>||	Click to move to the last page.	N/A	Go to last page.
17	List per page	Choose number of records displayed in 1 page.	N/A	Show number of program list per page
18	View program	Click to move to training program list	N/A	Navigate to view program page.
19	Create program	Click to move to create program page	N/A	Navigate to create training program

Table 49 - <User Interface-Button/Hyperlinks> Training Program List

5.4.2. Filter table

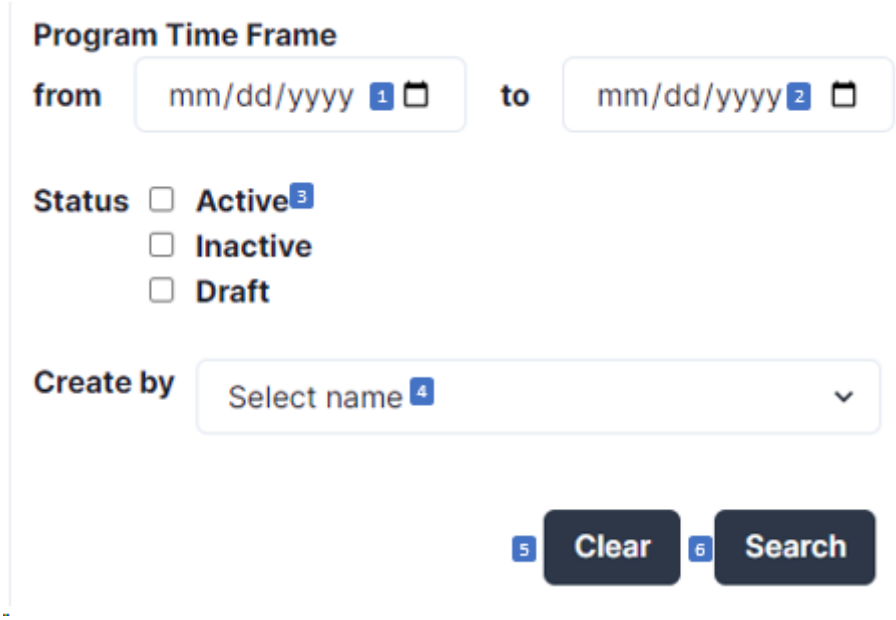


Figure 247 - Filter table UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	From	Select the day begin	No	Yes	Text box	Date time
2	To	Select the day end	No	Yes	Label	Date time
3	Status	Display 3 status: active, inactive, draft. Click to choose between those.	No	Yes	Check box	String

4	Create by	Select the name whose created that program	No	Yes	Single choice dropdown	String

Table 50 - <User Interface-Field> Filter table

Buttons/Hyperlinks:

No	Field Name	Description	Validation	Outcome
5	Clear	Clear all filter option	N/A	Clear all options successfully
6	Search	Click to search program with the option has been chosen	Validate of field required	Navigate to view program page to show the list has been filtered

Table 51 - <User Interface-Button/Hyperlinks> Filter table

5.4.3. Program Management

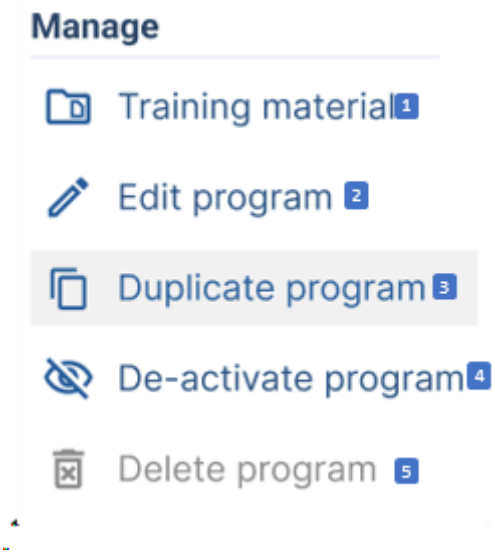


Figure 248 - Program Manage UI

Buttons/Hyperlinks:

No	Field Name	Description	Validation	Outcome
1	Training material	Show training material detail	N/A	Navigate to view program detail page
2	Edit syllabus	User update syllabus detail	N/A	Navigate to create training program page
3	Duplicate syllabus	Create a copy syllabus	N/A	Show success message.
4	De-activate	Change status of	N/A	Show success

	syllabus	syllabus		message.
5	Delete syllabus	Delete syllabus	Syllabus must be inactive	Show success message.

Table 52 - <User Interface-Button/Hyperlinks> Program Management

5.4.4. Create Training Program

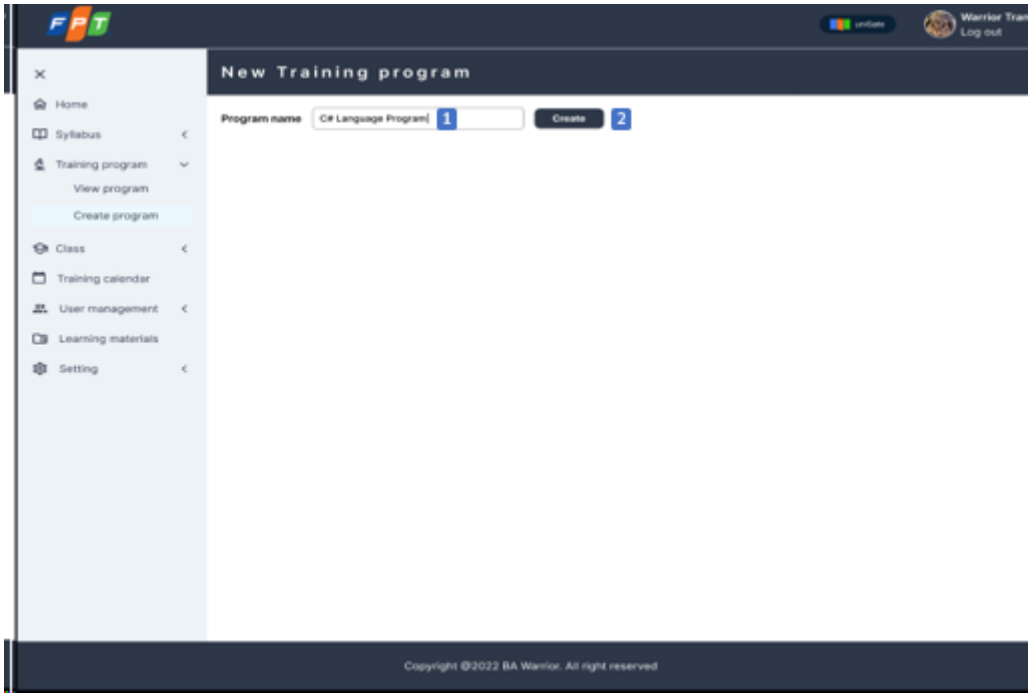


Figure 249 - Create Training Program UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Program name	Program name textbox	No	Yes	Text box	String

Table 53 - <User Interface-Field> Create Training Program

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
2	Create	Create name of the program	Validate field format	Move to next page user selects syllabus.

Table 54 - <User Interface-Button/Hyperlinks> Create Training Program

5.4.5. Add Syllabus

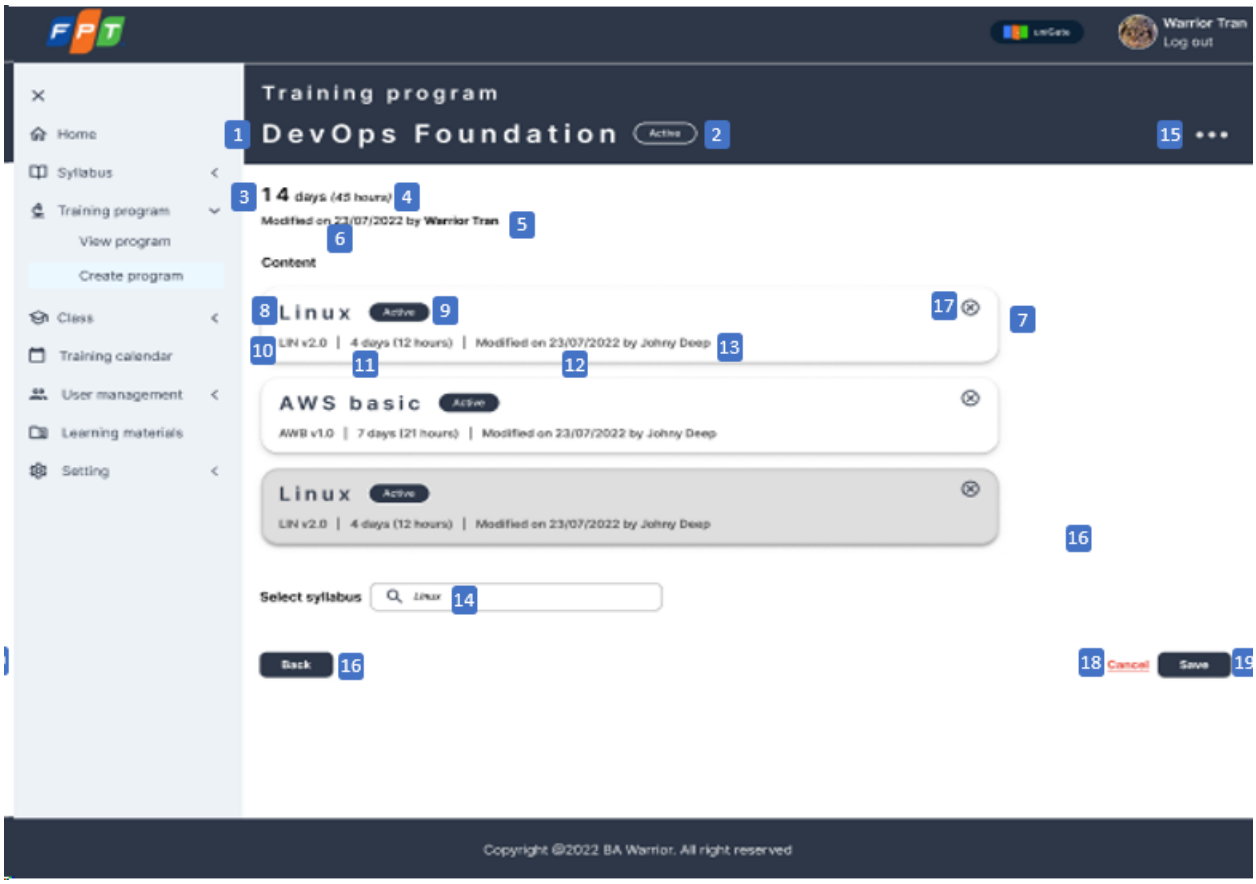


Figure 250 - Add syllabus UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Training program name	Training program name	No	Yes	Text box	String
2	Status	Training program status: active, inactive	Yes	Yes	Label	String
3	Training program total date	Training program total date	Yes	Yes	Label	String
4	Training program total duration	Training program total duration	Yes	Yes	Label	String
5	Training program last	Training program last	Yes	Yes	Label	String

	modified by	modified by				
6	Training program last modified date	Training program last modified date	Yes	Yes	Label	String
7	Syllabus card	Syllabus card	No	Yes	Tab	String
8	Syllabus name	Syllabus name	Yes	Yes	Label	String
9	Syllabus status	Syllabus status: active, inactive	Yes	Yes	Label	String
10	Syllabus code and version	Syllabus code and version	Yes	Yes	Label	String
11	Syllabus total date & duration	Syllabus total date & duration	Yes	Yes	Label	String
12	Syllabus last modified date	Syllabus last modified date	Yes	Yes	Label	String
13	Syllabus last modified by	Syllabus last modified by	Yes	Yes	Label	String
14	Select syllabus	Select syllabus from suggestion list	Yes	Yes	Text box	String

Table 55 - <User Interface-Field> Add syllabus

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
15	Manage menu button	Open manage menu	N/A	Navigate to training program manage
16	Back	Move to the create program name page	N/A	Navigate to alert popup
17	Delete card	Delete syllabus has been searched	N/A	Delete successfully
18	Cancel	Back to "List of Training Program" screen	N/A	Navigate to view program page
19	Save	Save training program to database with	Validate all fields	Navigate to view program page, show

		"inactive" status	required	success message.

Table 56 - <User Interface-Button/Hyperlinks> Add syllabus

5.4.6. Training program manage

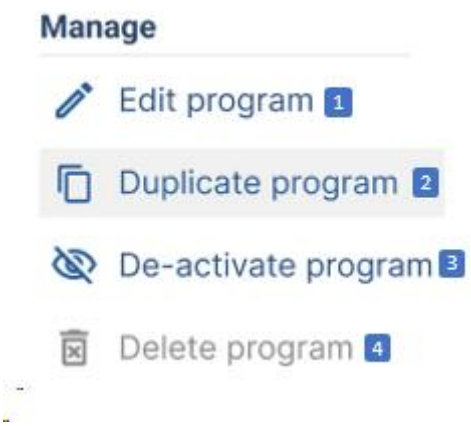


Figure 251 - Training program manage UI

Buttons/Hyperlinks:

No	Field Name	Description	Validation	Outcome
1	Edit program	User update program detail	N/A	Navigate to create training program page
2	Duplicate program	Create a copy program	N/A	Show success message.
3	De-activate program	Change status of program	N/A	Show success message.
4	Delete program	Delete program	Program must be inactive	Show success message.

Table 57 - <User Interface-Button/Hyperlinks> Training program manage

5.4.7. View program detail

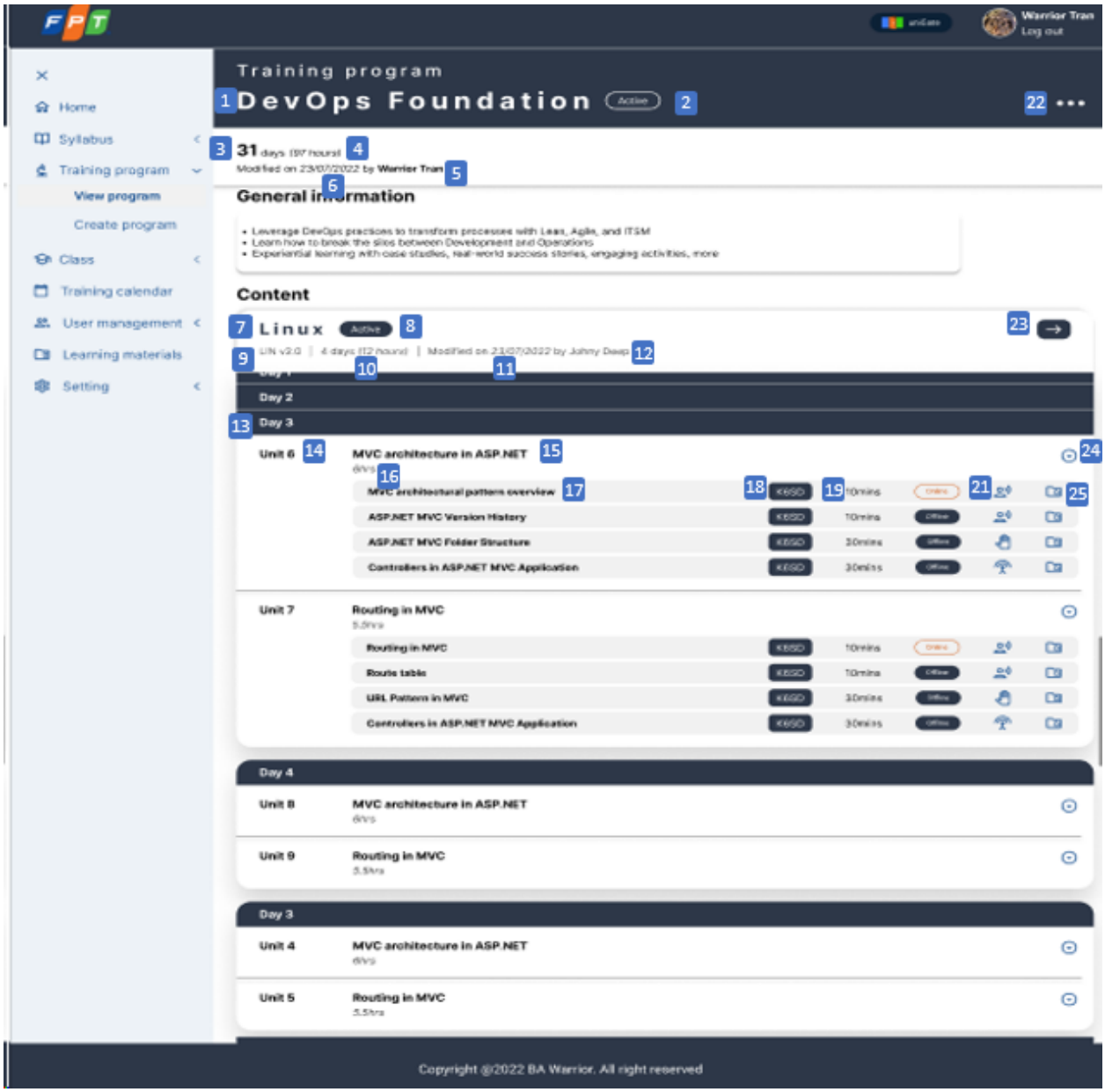


Figure 252 - View program detail UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Training program name	Training program name	No	Yes	Text box	String
2	Status	Training program status: active, inactive	Yes	Yes	Label	String

3	Training program total date	Training program total date	Yes	Yes	Label	String
4	Training program total duration	Training program total duration	Yes	Yes	Label	String
5	Training program last modified by	Training program last modified by	Yes	Yes	Label	String
6	Training program last modified date	Training program last modified date	Yes	Yes	Label	String
7	Syllabus name	Syllabus name	Yes	Yes	Label	String
8	Syllabus status	Syllabus status: active, inactive	Yes	Yes	Label	String
9	Syllabus code and version	Syllabus code and version	Yes	Yes	Label	String
10	Syllabus total date & duration	Syllabus total date & duration	Yes	Yes	Label	String
11	Syllabus last modified date	Syllabus last modified date	Yes	Yes	Label	String
12	Syllabus last modified by	Syllabus last modified by	Yes	Yes	Label	String
13	Syllabus day	Syllabus number: 1, 2, 3	Yes	Yes	Label	String
14	Unit number	Unit number: 1, 2, 3	Yes	Yes	Label	String
15	Unit name	Unit name	Yes	Yes	Label	String
16	Unit total duration	Unit total duration	Yes	Yes	Label	String
17	Chapter name	Chapter name	Yes	Yes	Label	String
18	Output standard	Output standard code	Yes	Yes	Label	String

	code					
19	Chapter duration	Chapter duration	Yes	Yes	Label	String
20	Chapter learning type	Chapter learning type: online, offline	Yes	Yes	Label	String
21	Chapter delivery type	Chapter delivery type	Yes	Yes	Label	String

Table 58 - <User Interface-Field> View program detail

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
22	Manage menu button	Open manage menu	N/A	Navigate to training program manage
23	Go to syllabus (arrow) button	Transit to syllabus detail page	N/A	Navigate to syllabus detail page
24	Collapse/ Expand	Collapse/ Expand syllabus section detail information (Item 39 -> 44)	N/A	
25	Training material	Training material of 1 single chapter	N/A	Navigate to training material detail

Table 59 - <User Interface-Button/Hyperlinks> View program detail

5.4.8. View program

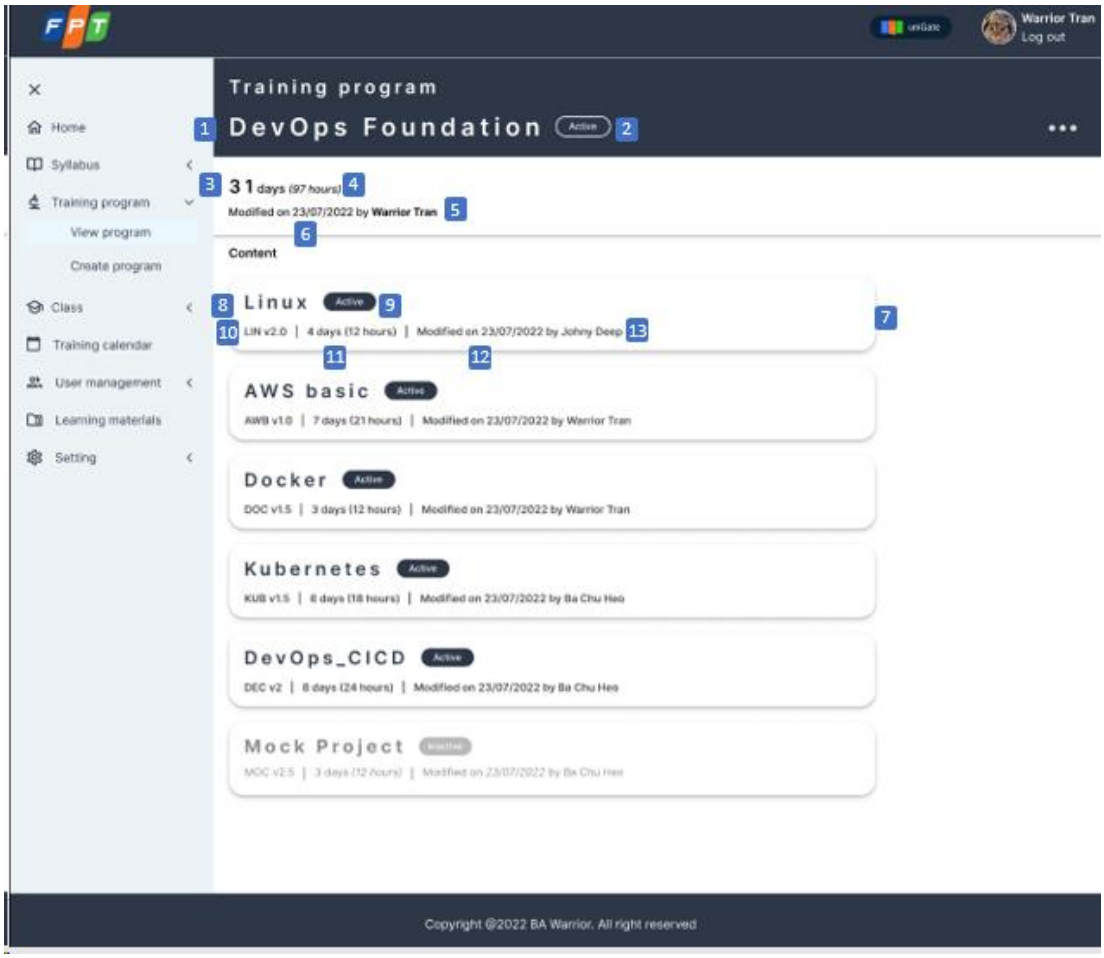


Figure 253 - View program UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Training program name	Training program name	No	Yes	Text box	String
2	Status	Training program status: active, inactive	Yes	Yes	Label	String
3	Training program total date	Training program total date	Yes	Yes	Label	String
4	Training program total duration	Training program total duration	Yes	Yes	Label	String

5	Training program last modified by	Training program last modified by	Yes	Yes	Label	String
6	Training program last modified date	Training program last modified date	Yes	Yes	Label	String
7	Syllabus card	Syllabus card	No	Yes	Tab	String
8	Syllabus name	Syllabus name	Yes	Yes	Label	String
9	Syllabus status	Syllabus status: active, inactive	Yes	Yes	Label	String
10	Syllabus code and version	Syllabus code and version	Yes	Yes	Label	String
11	Syllabus total date & duration	Syllabus total date & duration	Yes	Yes	Label	String
12	Syllabus last modified date	Syllabus last modified date	Yes	Yes	Label	String
13	Syllabus last modified by	Syllabus last modified by	Yes	Yes	Label	String

Table 60 - <User Interface-Field> View program

5.4.9. Training materials

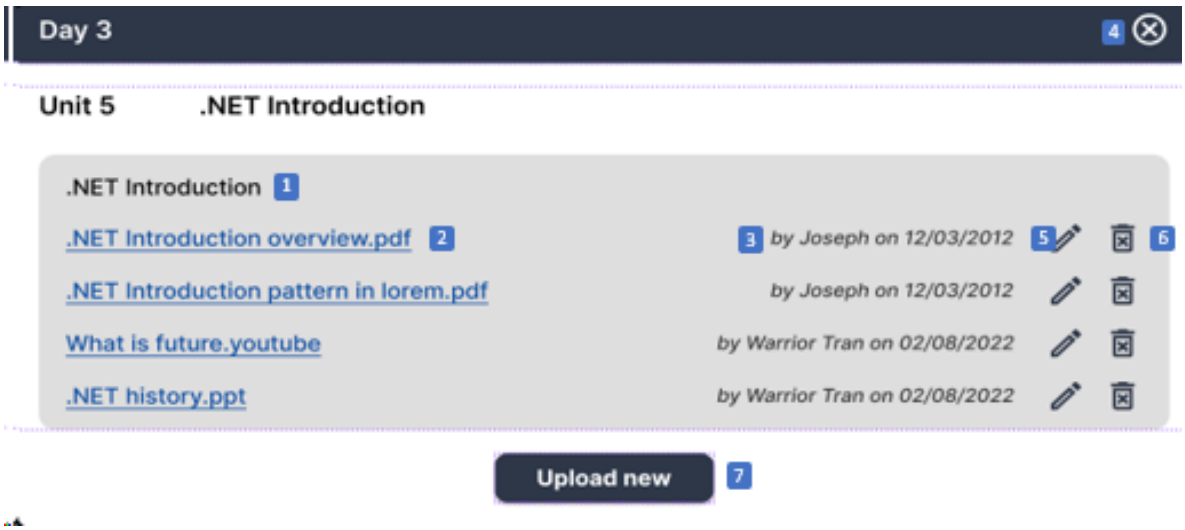


Figure 254 - Training materials UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Chapter name	Name of the chapter of those training material belong to	Yes	Yes	Label	String
2	Material name	Name of the file or link training material	No	Yes	Label	Hyper link
4	Modify information	Name of the file or link training material	Yes	Yes	Label	String

Table 61 - <User Interface-Field> Training materials

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Close	Close the training material popup	N/A	Close popup successfully.
2	Edit	Edit training materials	N/A	Navigate to user file explorer
3	Delete	Delete training material	N/A	Show success message.
4	Upload new	Upload new training material	N/A	Navigate to user file explorer.

Table 62 - <User Interface-Button/Hyperlinks> Training materials

5.4.10. Import Training Program

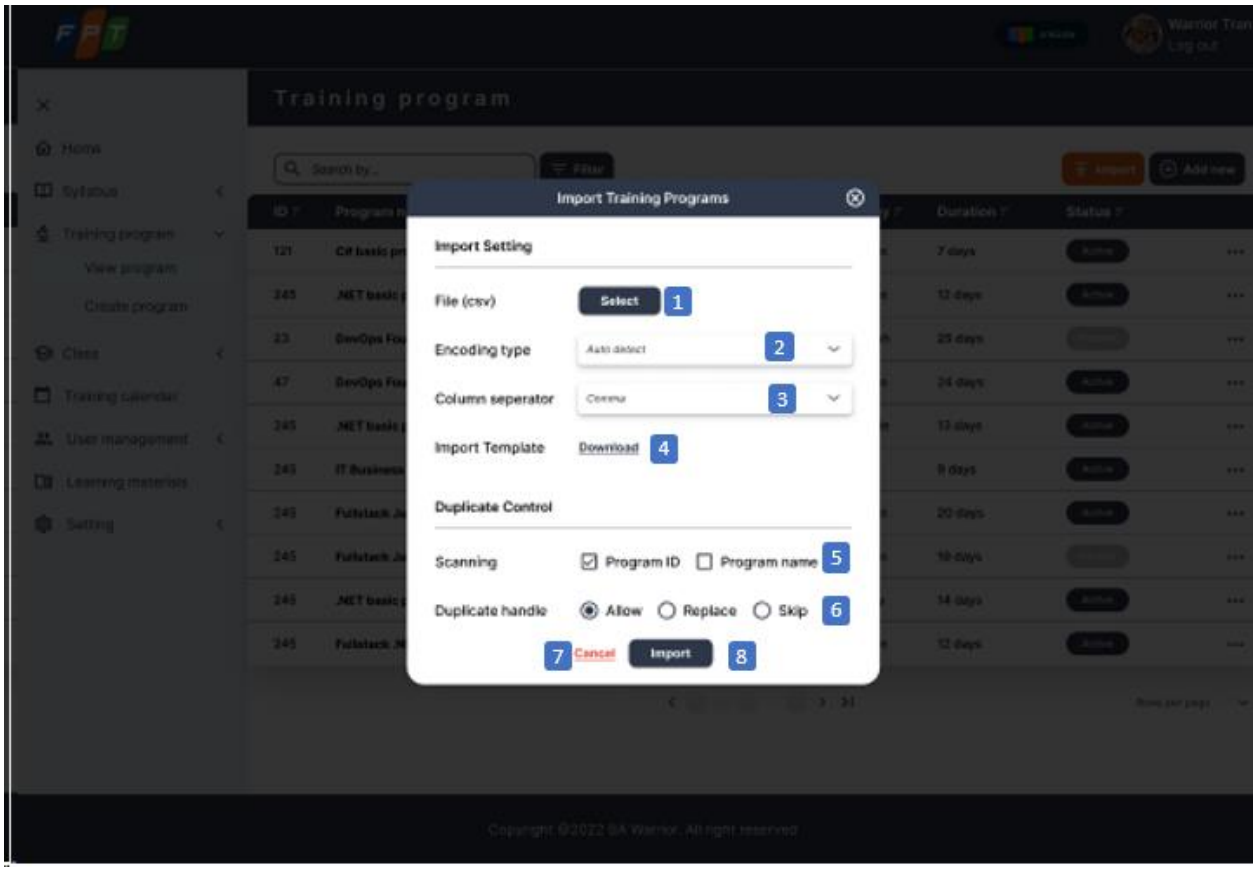


Figure 255 - Import Training Program UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
2	Encoding type	Encoding type of the file	No	Yes	Drop down list	String
3	Column separator	Column separator of the file	No	Yes	Drop down list	String
5	Scanning	Scanning the version and code of the file	No	Yes	Check box	String
6	Duplicate handle	Option to change when file duplicate	No	Yes	Radio check box	String

Table 63 - <User Interface-Field>Import Training Program

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Select	Select file to import	Only excel or csv file	Show file name has just been imported
4	Download	Download the import template	N/A	Navigate to user file explorer
7	Cancel	Cancel import process	N/A	Navigate to training program list
8	Import	User import new data	N/A	Show success message

Table 64 - <User Interface-Button/Hyperlinks>Import Training Program

5.5. Student Management Interface

5.5.1. Display Student List By Class

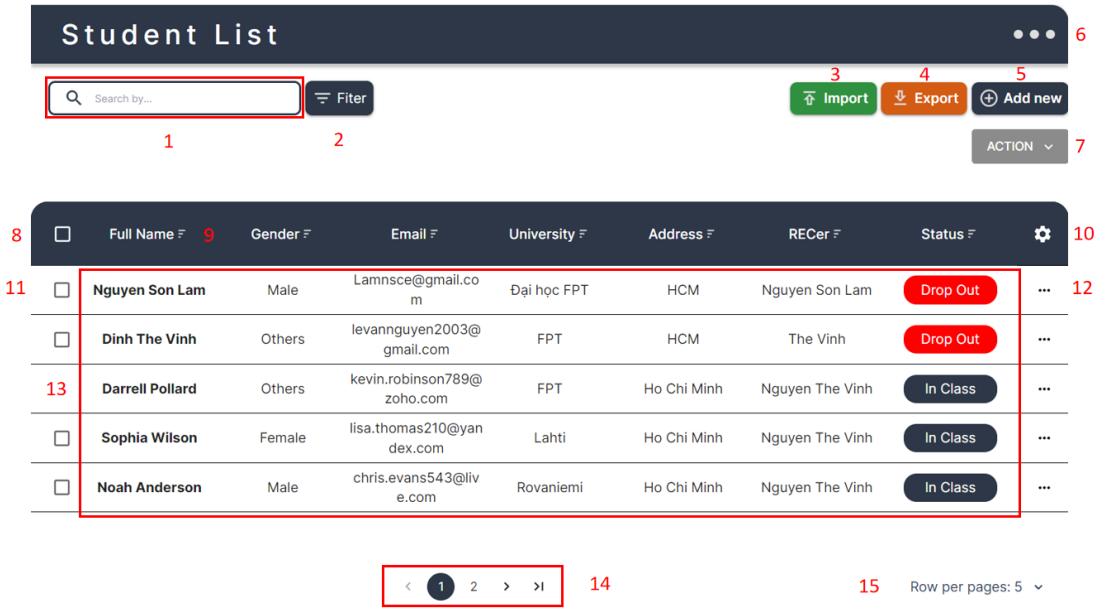


Figure 256 - Student list by class

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Search by	Search user by Full Name, Address, email, major.	No	No	TextBox	String	N/A

Table 65 - <User Interface-Field> Student list by class

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
2	Filter	Click to filter users.	N/A	Display filter popup
3	Import	Click to import users into class	N/A	Display Import popup
4	Export	Export users to .xlsx file	N/A	Download .xlsx file of list users
5	Add new	Add new student into class	N/A	Navigate to add new student screen
6		Click to collapse or expand the list of students.	N/A	Collapse or expand the list of student.
7	Action	Click to choose some action with student choosed.	N/A	Display menu for actions
8		Select all students	N/A	Select all students by check box
9		Sort by that field	N/A	Sort by that field sequential: asc, desc, normal.
10		Click to choose what columns that displayed	N/A	Open select columns popup
11		Select that student	N/A	Select that student by checkbox
12		Select action detail for that student	N/A	Display menu action detail.
13	Pagination	Pagination	N/A	Change page number.
14	Row per pages	Change row per pages	N/A	Change number of student display on table.

Table 66 - <User Interface-Button/Hyperlinks> Student list by class

5.5.2. Add Class Infor Popup

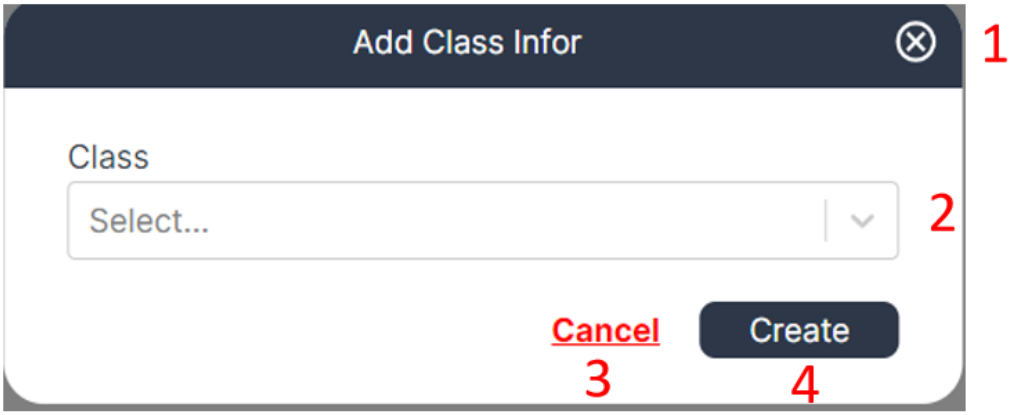


Figure 257 - Add Class Infor Popup

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
2	Class	Select a new class	No	Yes	Drop down	String	N/A

Table 67 - <User Interface-Field> Edit Student

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	⊗	Click to close popup	N/A	Close Popup
3	Cancel	Click to cancel add new class for student	N/A	Cancel add
4	Create	Click to add new class information for student	N/A	create

Table 68 - <User Interface-Button/Hyperlinks> Edit Student

5.5.3. Edit Student

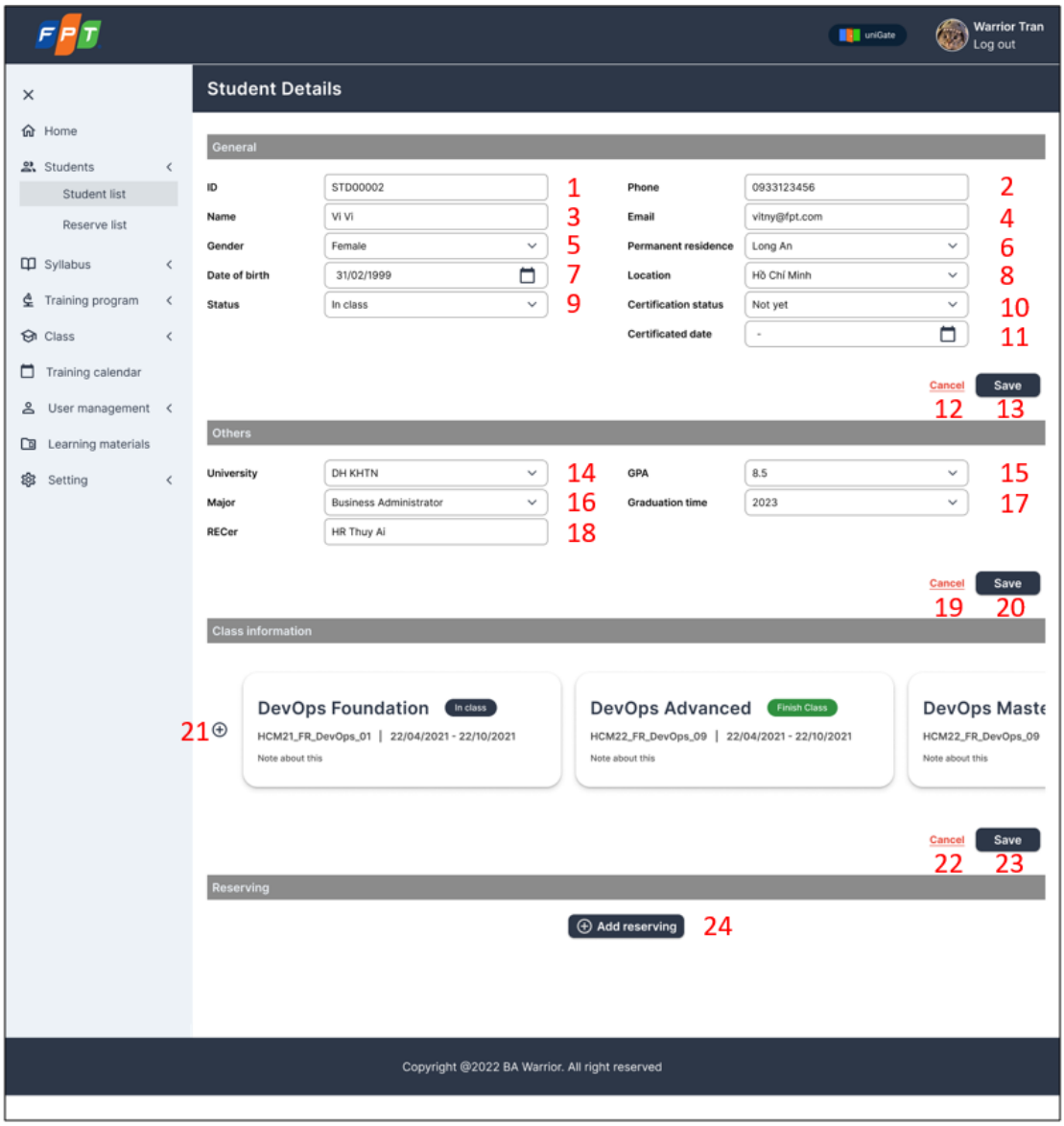


Figure 258 - Edit Student

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	ID	ID of student	No	Yes	Text Box	String	N/A
2	Phone	Phone number of user	No	Yes	Text Box	String	N/A
3	Name	Name of Student	No	Yes	Text Box	String	N/A
4	Email	Email of Student	No	Yes	Text	String	N/A

					Box		
5	Gender	Gender of Student	No	Yes	Drop Down	String	N/A
6	Permanent residence	Address of Student	No	Yes	Drop Down	String	N/A
7	Date of birth	Date of birth of Student	No	Yes	Date picker	Date time	N/A
8	Location	Location of Student	No	Yes	Drop Down	String	N/A
9	Status	Status of student in that class	No	Yes	Drop Down	String	N/A
10	Certificate status	Certificate status of student in that class	No	No	Drop Down	String	N/A
11	Certificate Date	Date get Certificate	No	No	Date Picker	Date Time	N/A
14	University	University of Student	No	Yes	Drop Down	String	N/A
15	GPA	GPA of Student	No	Yes	Drop Down	Numbe r	N/A
16	Major	Major of student	No	Yes	Drop Down	String	N/A
17	Graduation time	Graduation time	No	Yes	Drop Down	Year	N/A
18	RECer	RECer of Student	No	Yes	Text Box	String	N/A

Table 69 - <User Interface-Field> Edit Student

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
12	Cancel	Click to cancel update student's infor	N/A	Cancel update
13	Save	Click to save new information	N/A	Save update
19		Add new class information for that student	N/A	Display add new class infor popup
5		Add new reserving for that student	N/A	Display add new reserving popup

Table 70 - <User Interface-Button/Hyperlinks> Edit Student

5.5.4. Import function

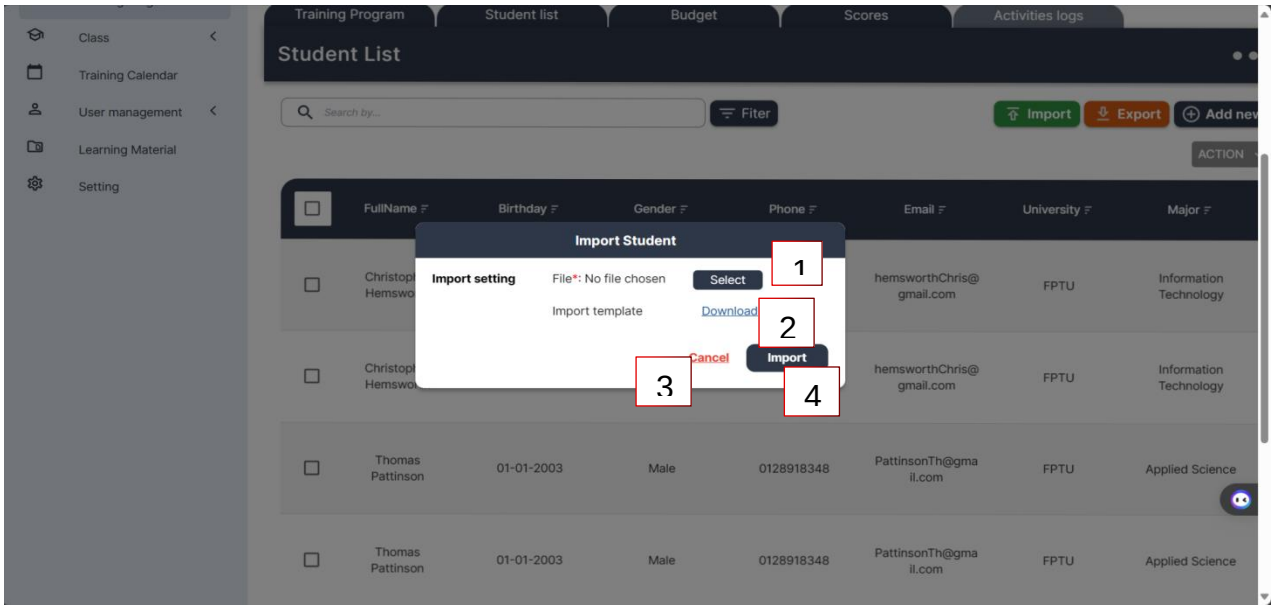


Figure 259 - Import 1 function UI

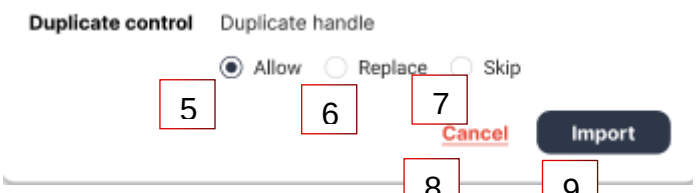


Figure 260 - Import 2 function UI

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Select	Choose a file from User desktop	N/A	Import popup
2	Download	Click to download a template file	N/A	Import popup
3	Cancel	Turn to Student List page	N/A	Student List page
4	Import	Click to import file into system	Check exist student	Import popup
5	Allow	Allow import student list exist	N/A	Import popup
6	Replace	Replace Student List Exist	N/A	Import popup

7	Skip	Skip Student List Exist	N/A	Import popup
8	Cancel	Back to Import popup 1	N/A	Import popup
9	Import	import student list exist	N/A	Student List page

Table 71 - <User Interface-Button/Hyperlinks> Import function

5.5.5. View system page

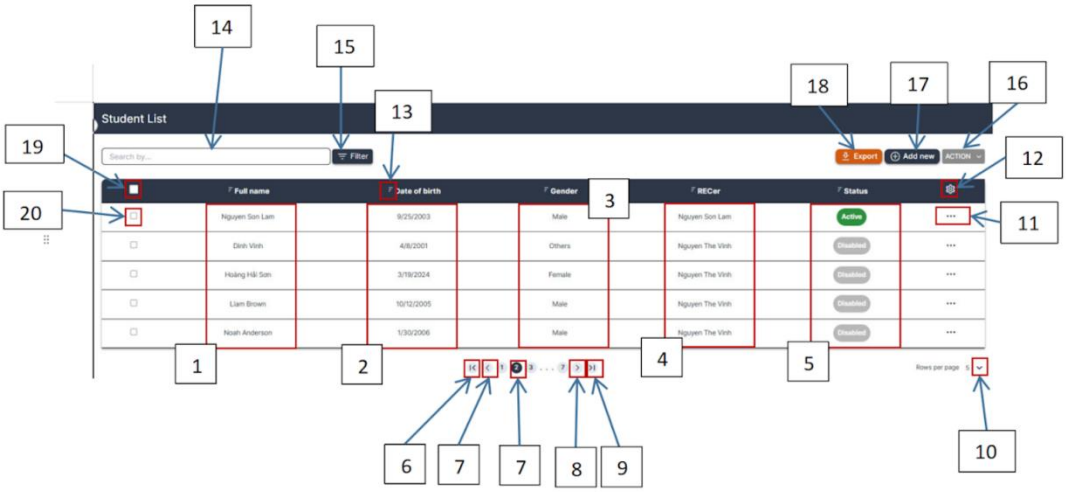


Figure 261 - View system page

Fields

No	Field Name	Description	Data Type
1	Full Name	Column showing the student's full name	String
2	Date of birth	Column showing the student's date of birth	Date
3	Gender	Column showing the student's gender	String
4	Recer	Column showing the student's Recer	String
5	Status	Column showing the student's status	String

Table 72 - <User Interface-Field>View system page

Buttons/Hyperlinks

No	Button Name	Description
6	Previous page	Button to navigate to the previous page

7	Page number	Button to navigate to a specific page
8	Next Page	Button to navigate to the next page
9	Last page	Button to navigate to the last page
10	Row per page	Drop down to choose how many student to display at once [EV6]
11	3 dot button	Drop down to interact with student [EV9]
12	Gear button	Button to open column pop-up to choose what column to display [EV8]
13	Sort button	Button to sort table based on what column the user is clicking on [EV5]
14	Search box	Text box to search for student based on full name, major, location [EV1]
15	Filter button	Button to open filter pop-up [EV7]
16	Action button	Button to interact with a batch of selected students at once [EV4]
17	Add new	Button to navigate to the add new student page [EV3]
18	Export	Button to export the current table to a xlsx file [EV2]
19	Select all	Button to select all student to manage
20	Check box	Target each student

Table 73 - <User Interface-Button/Hyperlinks> View system page

5.5.6. Student management pop-up

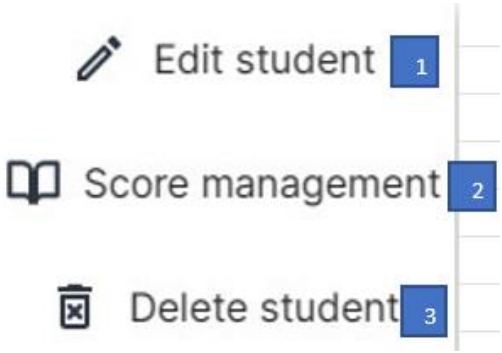


Figure 262 - Student management pop-up

Buttons/Hyperlinks

No	Button Name	Description
1	Edit student	Edit the student information
2	Score management	Manage the score of the student
3	Delete student	Open a pop-up to confirm deleting the student

Table 74 - <User Interface-Button/Hyperlinks> Student management pop-up

5.5.7. Confirm delete pop-up

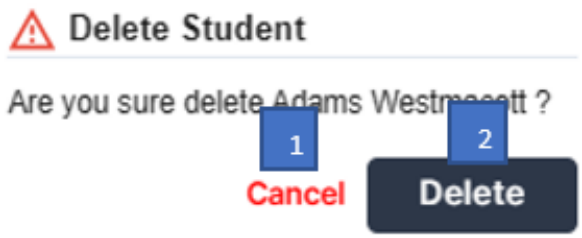


Figure 263 - Confirm delete pop-up

Buttons/Hyperlinks

No	Button Name	Description
1	Cancel	Cancel the deletion and close the pop-up
2	Delete	Change the corresponding student's status to disabled

Table 75 - <User Interface-Button/Hyperlinks> Confirm delete pop-up

5.5.8. Select column pop-up

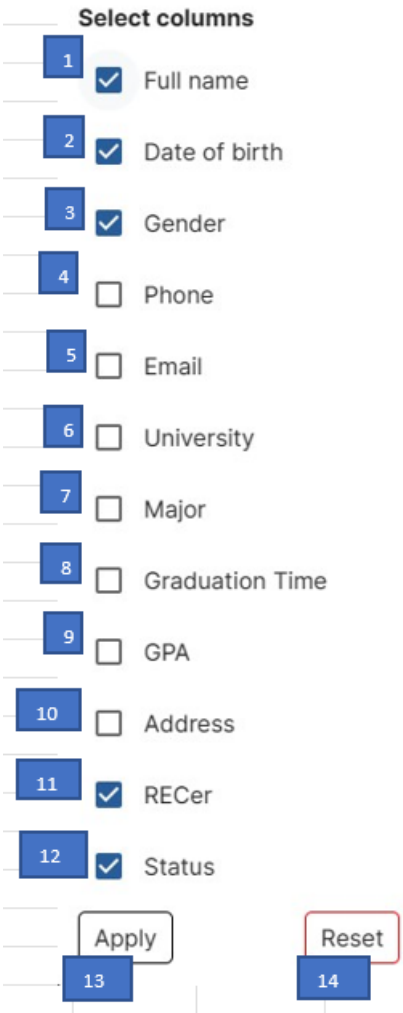


Figure 264 - Select column pop-up

Buttons/Hyperlinks

No	Button Name	Description
1	Full name	Select the full name column to display
2	Date of birth	Select the date of birth column to display
3	Gender	Select the gender column to display
4	Phone	Select the phone column to display
5	Email	Select the email column to display

6	University	Select the university column to display
7	Major	Select the major column to display
8	Graduation Time	Select the graduation time column to display
9	GPA	Select the GPA column to display
10	Address	Select the address column to display
11	Recer	Select the recer column to display
12	Status	Select the status column to display
13	Apply	Apply the column selection
14	Reset	Change the column selection to the default setitngs

Table 76 - <User Interface-Button/Hyperlinks> Select column pop-up

5.5.9. Filter pop-up

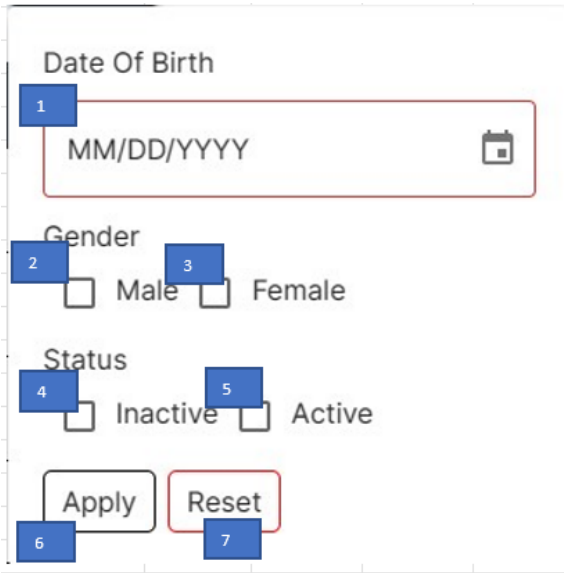


Figure 265 - Filter pop-up

Buttons/Hyperlinks

No	Button Name	Description
1	Date of birth	Add a date of birth filter option
2	Male	Add male gender filter option
3	Female	Add female gender filter option
4	Inactive	Add inactive status filter option
5	Active	Add active status filter option
6	Apply	Confirm the filter settings
7	Reset	Remove all filter options

Table 77 - <User Interface-Button/Hyperlinks> Filter pop-up

5.5.10. Action pop-up

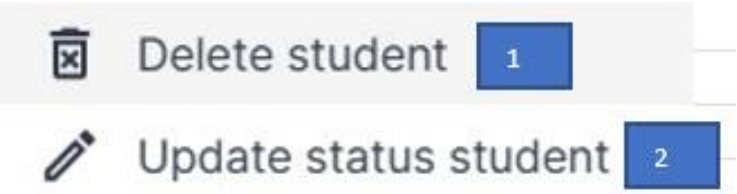


Figure 266 - Action pop-up

Buttons/Hyperlinks

No	Button Name	Description
1	Delete student	Delete the selected students
2	Update student status	Change status of the selected students

Table 78 - <User Interface-Button/Hyperlinks> Action pop-up

5.5.11. Add student page

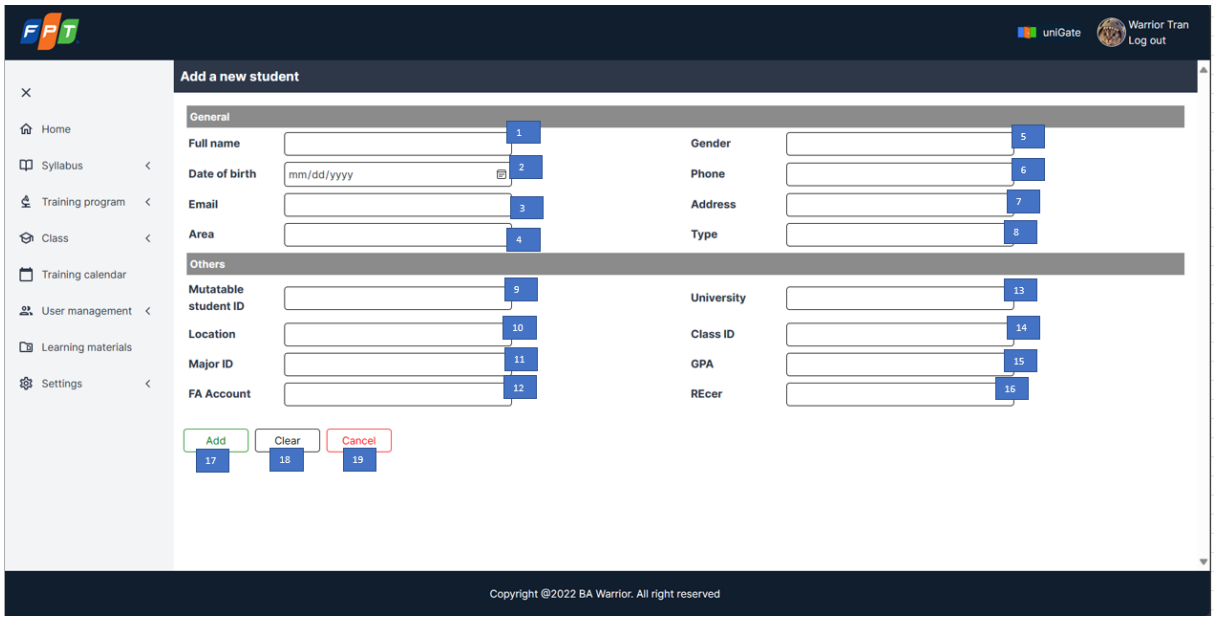


Figure 267 - Add student page

Fields

No	Field Name	Description	Data Type
1	Full Name	Input field for the full name of the student	String
2	Date of birth	Input field for the date of birth of the student	Date
3	Email	Input field for the email of the student	String
4	Area	Input field for the area of the student	Number
5	Gender	Input field for the gender of the student	String
6	Phone	Input field for the phone of the student	String
7	Address	Input field for the address of the student	String
8	Type	Input field for the type of the student	Date
9	Mutatable student ID	Input field for the mutatable student ID of the student	Number
10	Location	Input field for the location of the student	String

11	Major ID	Input field for the major ID of the student	String
12	FA Account	Input field for the fa account of the student	String
13	University	Input field for the university of the student	String
14	Class ID	Input field for the class ID of the student	String
15	GPA	Input field for the GPA of the student	String
16	Recer	Input field for the Recer of the student	String

Table 79 - <User Interface-Field> Add student page

Buttons/Hyperlinks

No	Button Name	Description
17	Add	Add the new student with the inputted information
18	Clear	Clear all the input field
19	Cancel	Cancel creation and return to the page prior to accessing the add student page

Table 80 - <User Interface-Button/Hyperlinks> Add student page

5.5.12. Edit student from System

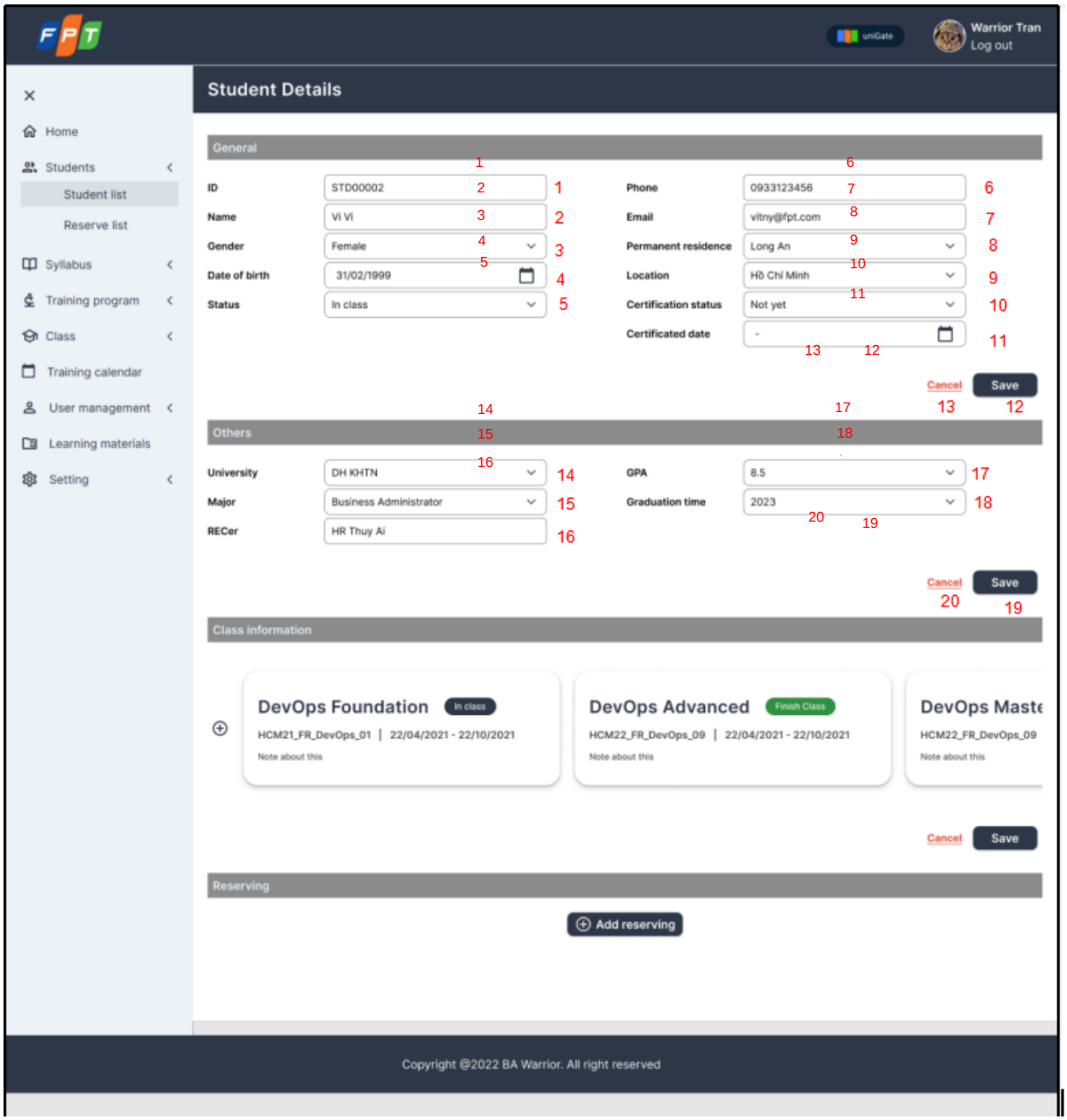


Figure 268 - Edit student from System

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Id	Display id of student	Yes	Yes	TextBox	String	N/A
2	Name	Display name of student to	No	Yes	TextBox	String	N/A

		change					
3	Gender	Display gender of student to change	No	Yes	ComboBox	String	N/A
4	Date of Birth	Display birthday of student to change	No	Yes	DateTimepicker	DateTime	N/A
5	Status	Display status of student to change	No	Yes	ComboBox	String	N/A
6	Phone	Display phone number of student to change	No	Yes	TextBox	String	N/A
7	Email	Display email address of student to change	No	Yes	TextBox	String	N/A
8	Permanent Residence	Display temporary address of student to change	No	Yes	ComboBox	String	N/A
9	Location	Display recent address of student to change	No	Yes	ComboBox	String	N/A
10	Certification Status	Display certification status of student to change	No	Yes	ComboBox	Int	N/A
11	Certification Date	Display certification date of student to change	No	Yes	DateTimePicker	Datetime	N/A
14	University	Display university of student to change	No	Yes	ComboBox	String	N/A
15	Major	Display major of student to	No	Yes	ComboBox	String	N/A

		change					
16	GPA	Display gpa of student to change	No	Yes	ComboBox	Decimal	N/A
17	Graduation Time	Display graduation time of student to change	No	Yes	DateTimePicker	DateTime	N/A

Table 81 - <User Interface-Field> Edit student from system

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
12	Save	Click to save the change.	N/A	None
13	Cancel	Don't save change and return to student detail page.	N/A	None
19	Save	Click to save the change.	N/A	None
20	Cancel	Don't save change and return to student detail page.	N/A	None

Table 82 - <User Interface-Button/Hyperlinks> Edit student from system

5.5.13. Edit Student Status in batch_Class.

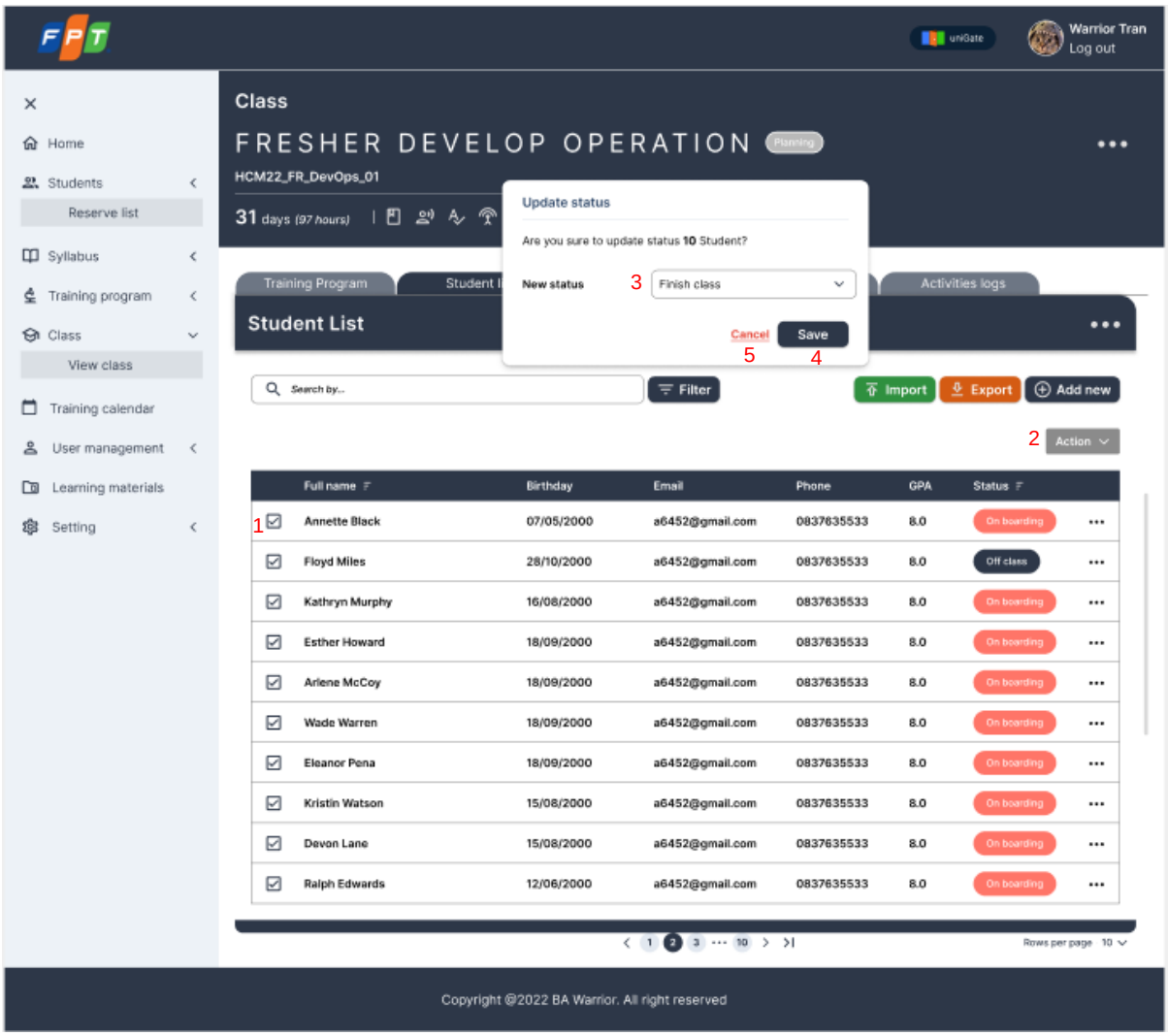


Figure 269 - Edit Student Status in batch_Class

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
3	New Status	Apply new status for all student selected	No	Yes	ComboBox	String	N/A
1	CheckBox	Check to choose number of student.	No	No	CheckBox	None	N/A

Table 83 - <User Interface-Field> Edit Student Status in batch_Class

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
2	Action List	Click to drop-down list of action.	N/A	None
4	Save	Save the change of status in batch.	N/A	None
5	Cancel	Ignore the change and exit from pop-up.	N/A	None

Table 84 - <User Interface-Button/Hyperlinks> Edit Student Status in batch_Class

5.6. Score Management Interface

5.6.1. Import Score

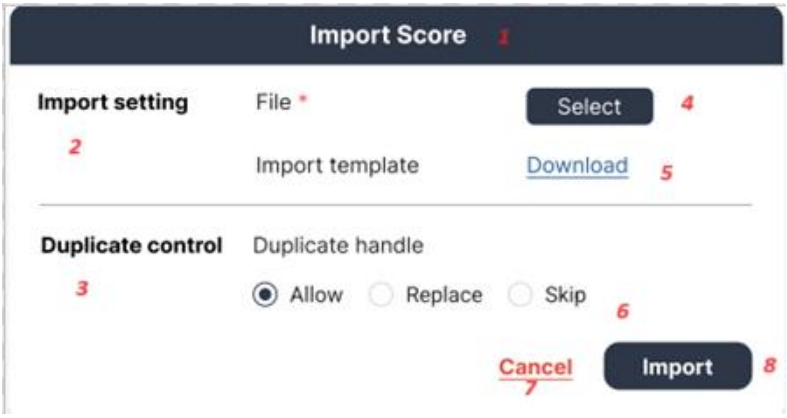


Figure 270 - Import Score

Fields

No.	Screen element	Format Type	Require	Default show	Data type	Data table	Default value	Description
1	Import Score	Header	Yes	Yes	N/A	-	N/A	Title of popup importing trainee's score
2	Import setting	Label	Yes	Yes	N/A	-	N/A	Title of section that allow user to have option to download excel template and/ or upload the excel file
3	Duplicate control	Label	Yes	Yes	N/A	-	N/A	Title of section that allow user to select

								one from 3 options : Allow/ Replace/ Skip
4	Select	Button	Yes	Yes	N/A	-	N/A	Allow user to upload the excel file
5	Download	Button	Yes	Yes	N/A	-	N/A	Allow user to download excel template ScoreImportTemplate
6	Allow/ Replace/ Skip	Radio button	Yes	Yes	N/A	-	N/A	Allow user to select one from 3 options: Allow/ Replace/ Skip
7	Cancel	Button	Yes	Yes	N/A	-	N/A	Allow user to cancel from the popup
8	Import	Button	Yes	Yes	N/A	-	N/A	Allow user to import the excel file on system

Table 85 - <User Interface-Field> Import Score

5.6.2. View the trainee's score, ranking and final assessment of all modules per class

Figure 271 - View the trainee's score, ranking and final assessment of all modules per class

Fields

No.	Screen element	Format Type	Require	Default show	Data type	Data table	Default value	Description
1	Full name	Label	Yes	No	Alphabetic	-	N/A	Name of trainee in a class
2	Account	Label	Yes	No	Alphabetic	-	N/A	Account of trainee in a class, this must be unique
3	Quiz	Label	Yes	No	Numeric	-	N/A	Quiz score of trainees in a class. Quizz score is taken from data input by admin.
4	ASM	Label	Yes	No	Numeric	-	N/A	ASM score of trainees in a class
5	Quizz final	Label	Yes	No	Numeric	-	N/A	Quiz final score of trainees in a class. Quiz final score is taken from data input by admin.
6	Audit	Label	Yes	No	Numeric	-	N/A	Audit score of trainees in a class. Audit score is taken from data input by admin.
7	Practice final	Label	Yes	No	Numeric	-	N/A	Final practice score of trainees in a class. Practice final score is taken from data input by admin.
8	Final module	Label	Yes	No	Numeric	-	N/A	Final score of trainees in a class.

								Final score is calculated upon the defined formular
9	GPA module	Label	N/A	No	Numeric	-	N/A	GPA score of trainees in a class. GPA score is calculated upon the defined formular
10	Level module	Label	N/A	No	Numeric	-	N/A	Level score of trainees in a class
11	Status	Icon	N/A	No	Alphabetic	-	N/A	Status of a trainees in a class. It must be FAILED or PASSED. If the GPA >= 60%, status is PASSED. Else, status is FAILED.
12	Import	Button	N/A	No	N/A	-	N/A	Allow user to import the excel file on system
13	Export	Button	N/A	No	N/A	-	N/A	Export file score

Table 86 - <User Interface-Field> View the trainee's score, ranking and final assessment of all modules per class

5.6.3. View final studying result per trainee

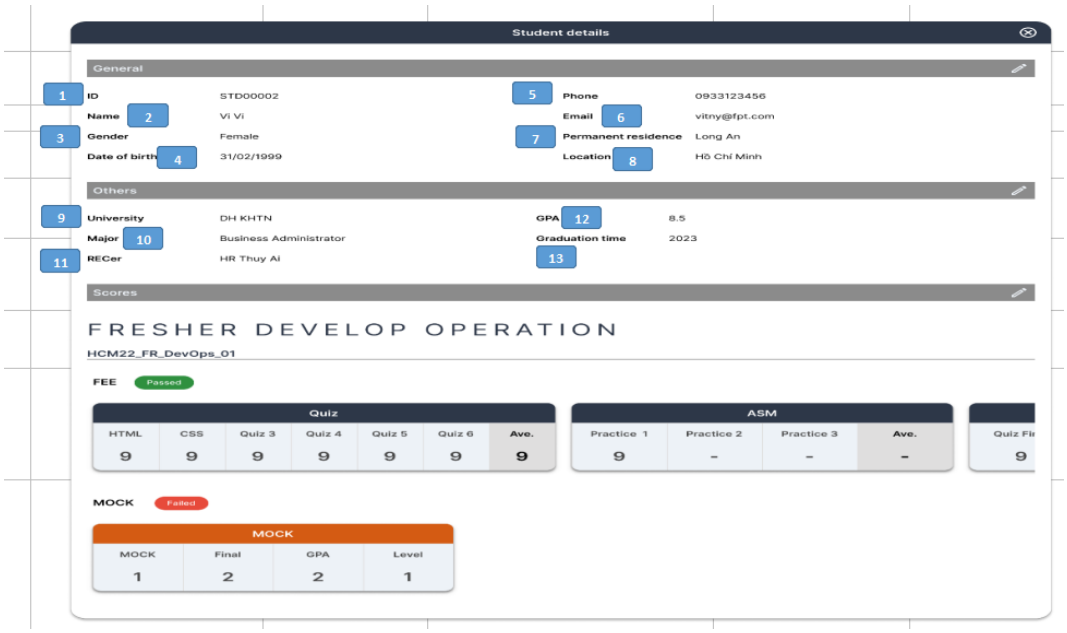


Figure 272 - View final studying result per trainee

Fields

No.	Screen element	Format Type	Require	Default show	Data type	Data table	Default value	Description
1	ID	Label	Yes	Yes	Numeric	-	N/A	ID of a trainee in specific class
2	Name	Label	Yes	Yes	Alphabetic	-	N/A	Name of the trainee
3	Gender	Label	Yes	Yes	Boolean	-	N/A	Gender of the trainee
4	Date of birth	Label	Yes	Yes	DateTime	-	N/A	Date of birth of the trainee
5	Phone	Label	Yes	Yes	Numeric	-	N/A	Phone of the trainee
6	Email	Label	Yes	Yes	Alphanumeric	-	N/A	Email of the trainee
7	Permanent residence	Label	No	Yes	Alphanumeric	-	N/A	Permanent residence of

								trainee
8	Location	Label	No	Yes	Alphanumeric	-	N/A	Location of the trainee

Table 87 - <User Interface-Field> View final studying result per trainee

5.6.4. Update the score list for each student

Figure 273 - Update the score list for each student

Fields

No.	Screen element	Format Type	Require	Default show	Data type	Data table	Default value	Description
1	Header	Text	Yes	Yes	Text	N/A	N/A	Display the popup title as "Scores"
2	Class name	Label	Yes	Yes	N/A	N/A	N/A	Display class name
3	Class ID	Label	Yes	Yes	N/A	N/A	N/A	Display class ID
4	Training program name + status	Label	N/A	N/A	N/A	Training_program.name,	N/A	- Display 2 modules as follows: + FEE has card type:

	(each card)					Training_program.status		Quiz, ASM, Quiz Final, Audit, Pratice Final, Final Module, GPA Module, Level Module, Status + MOCK has card type: MOCK -Staus: Pass/ Fail (Refer to Mark Tracking)
5	Card	Textbox	Yes	N/A	Plus number	N/A		Input points into each cell on the card
6	Cancel	button	N/A	Yes	N/A	-	N/A	Cancel editing Scores information and return Student details in Score list of a class
7	Save	button	N/A	Yes	N/A	-	N/A	Allow user to save inputted information, if all the information is valid then appears message inform "The scores have been successfully updated."

Table 88 - <User Interface-Field> Update the score list for each student

5.6.5. Update status of certificate award

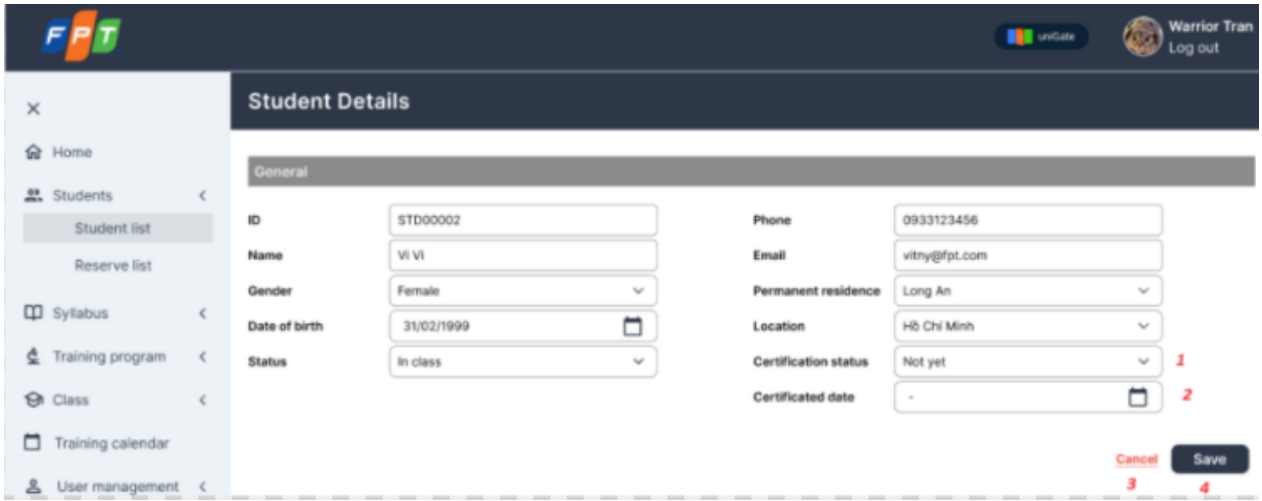


Figure 274 - Update status of certificate award

Fields

No.	Screen element	Format Type	Require	Default show	Data type	Data table	Default value	Description
1	Certificate Status	Dropdown list	Yes	N/A	Dropdown list	-	N/A	Award status contains: **- Done**: When the admin has awarded the certificate to the student, select status is "Done" **- Not yet**: When the admin has not yet awarded the certificate to the student, select status is "Not yet"
2	Certificate date	Date picker	Yes	N/A	Date	-	N/A	- Only enable when Certificate Status is **Not yet** - Admin chooses the date to award the certificate - When Certificate Status is **Done**, Certificate date will automatically be blank and disabled
3	Cancel	button	N/A	Yes	N/A	-	N/A	Cancel editing Certificate information and return Student list screen
4	Save	button	N/A	Yes	N/A	-	N/A	Allow user to save inputted

								information, if all of the information is valid then appears message inform edit successfully.

Table 89 - <User Interface-Field> Update status of certificate award

5.7. Reservation Management Interface

5.7.1. Reserve List Page

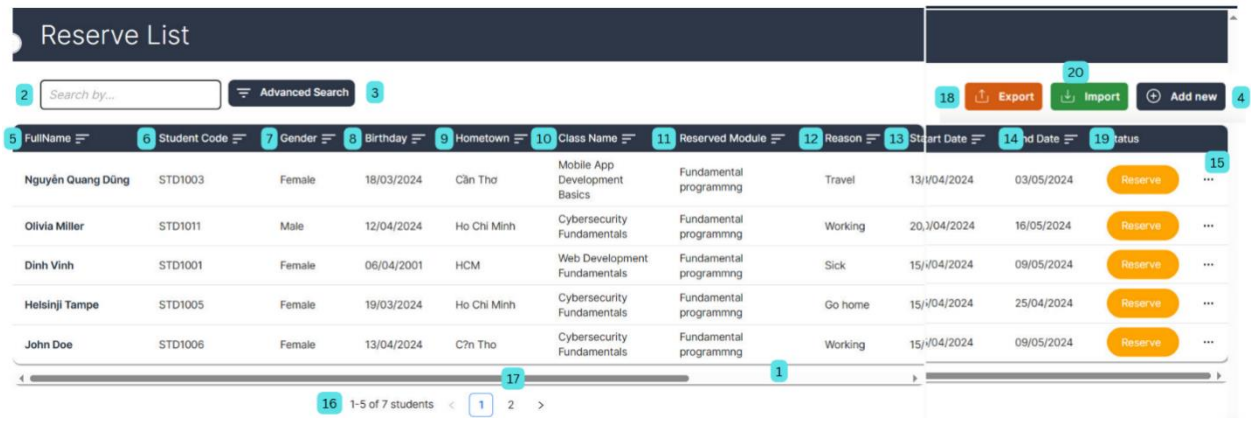


Figure 275 - Reserve List Page

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Reserve list	List of reserved students	Yes	No	Table	List	N/A
2	Quick search	- Allow user input data and search as auto-completed search. Search results will be filtered & displayed as input letters. - Searched by Student code, Full name, Email - Searched conditions will be displayed right under the quick search textbox, as tag with grey color & “x” icon for deleting	No	No	TextBox	String	N/A

		- Display maximum 4 tags (for 3 conditions: ID, Full name, Email)					
5	Full name	Full name of the students is displayed.	Yes	No	Text	String	N/A
6	Student code	Student code of the students is displayed.	Yes	No	Text	String	N/A
7	Gender	Gender of the students is displayed.	Yes	No	Text	String	N/A
8	Birthday	Birthday of the students is displayed.	Yes	No	Text	String	N/A
9	Hometown	Hometown of the students is displayed.	Yes	No	Text	String	N/A
10	Class	Name of the class is displayed.	Yes	No	Text	String	N/A
11	Reserved module	Name of the reserved module is displayed.	Yes	No	Text	String	N/A
12	Reason	Reserve reason is displayed.	Yes	No	Text	String	N/A
13	Start Date	Start Date of the period of reservation is displayed.	Yes	No	Text	String	N/A
14	End Date	End Date of the period of reservation is displayed.	Yes	No	Text	String	N/A
16	Total student	Total students of the list are displayed.	Yes	No	Text	String	N/A
19	Status	Status of student is displayed (Reserve)	Yes	No	Text	String	N/A

Table 90 - <User Interface-Field> Reserve List Page

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	Advanced search	Click to search all inputted fields	N/A	Return a new list in the "Reserve list" Page.
4	Add new	Click to add new reserved student	N/A	Navigate to the "Add New Student" Page.
15	Pop-up menu	Onclick to display pop-up menu - Re-class: on click to change the status to "In class". - Send Email: on click to send	N/A	Pop up the Sub-action menu.

		reminder to student when reserve period is nearly over - Drop class: on click to change status the status of attending to "Drop out" - Remove reserve: on click to remove the status "Reserved".		
17	Page count	Click to change page of list.	N/A	Change page of list.
18	Export	Click to export data of current page to excel	N/A	Return data of current page to excel file
20	Export	User export data	N/A	Show success message

Table 91 - <User Interface-Button/Hyperlinks> Reserve List Page

5.7.2. Add New Reserved Student Page

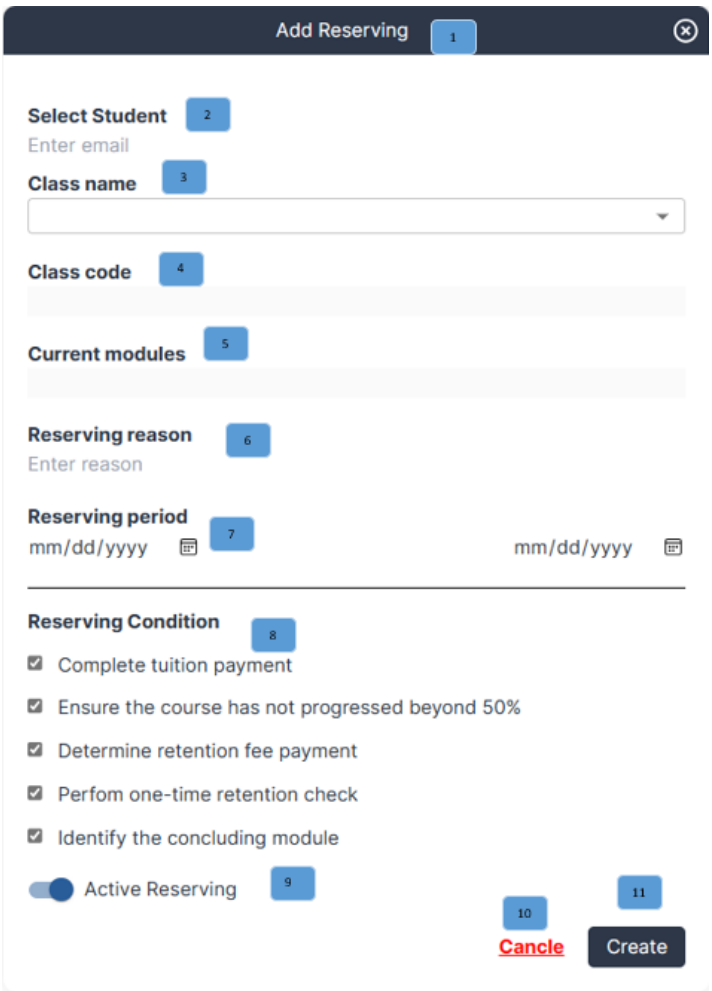


Figure 276 - Add New Reserved Student Page

Fields:

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Add reserving	Fixed word	Yes	No	Text	String	N/A
2	Select student	Allow users to input the information. Update the information shown in default if it is incorrect	No	Yes	TextBox	String	N/A
3	Class name	Name of the class is displayed for user choosing.	No	Yes	Select Box	String	N/A
4	Class code	Class code is displayed.	Yes	Yes	TextBox	String	N/A
5	Current modules	Current module of class is displayed.	Yes	Yes	TextBox	String	N/A
6	Reserving reason	Allow users to input the reserving reason.	No	Yes	TextBox	String	N/A
7	Reserving period	Start date and end date are displayed (format: dd/mm/yyyy). Reserving period is allowed maximum within 6 months.	No	Yes	Date	Date time	N/A
8	Reserving conditions	Display list of options: - Complete tuition payment - Ensure the course has not progressed beyond 50% - Determine retention fee payment - Perform one-time retention check Identify the concluding module	No	Yes	Checkbox	N/A	N/A
9	Activate reserving	Allow the user to switch on/ off the toggle button to change the attending status to reserving mode.	No	Yes	Toggle button	N/A	N/A

Table 92 - <User Interface-Field> Add New Reserved Student Page

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Cancel	Click to cancel add new reserved student	N/A	Return a new list in the "Reserve list" Page.
2	Create	Click to add new reserved student	N/A	Pop up "Add new student successfully"

Table 93 - <User Interface-Button/Hyperlinks> Add New Reserved Student Page

5.7.3. ReClass Dialog

Figure 277 - ReClass Dialog

Fields:

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Reserving detail	Fixed word	Yes	No	Text	String	N/A
2	Dev 1	Display Reserved Class Name of student	Yes	No	Text	String	N/A
3	CSI101	Display Reserved Class Code of student	Yes	No	Text	String	N/A
4	Class Period	Display Period of student from start date to latest studied date before leave	Yes	No	Text	String	N/A
5	Student Score	Fixed word	Yes	No	Text	String	N/A
6	Current Module	Fixed word	Yes	No	Text	String	N/A
7	Java	Display latest joined module	Yes	No	Text	String	N/A
8	FEE	Fixed word	Yes	No	Text	String	N/A
9	Quiz	Display Quiz until the date reserve	Yes	No	Table	List	N/A
10	ASM	Display Assignment until the date reserve	Yes	No	Table	List	N/A
11	Mock	Fixed word	Yes	No	Text	String	N/A
12	Mock Table	Display Mock until the date reserve	Yes	No	Table	List	N/A
13	Reserving information	Fixed word	Yes	No	Text	String	N/A
14	Period	Display Period of student from start date to latest studied date before leave	Yes	No	Text	String	N/A
15	Reason	Display reason to	Yes	No	Text	String	N/A

		reserve					
16	Conditions	Display checked boxed to ensure meet condition of reserve before	Yes	No	Checkboxes	N/A	N/A
17	Reclass Possibility	Fixed word	Yes	No	Text	String	N/A

Table 94 - <User Interface-Field> ReClass Dialog

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Cards of class	Click to view ReClass Recommendation of this Re Class.	N/A	Navigate to ReClass Recommendation of this class.

Table 95 - <User Interface-Button/Hyperlinks> ReClass Dialog

5.7.4. ReClass Recommendation

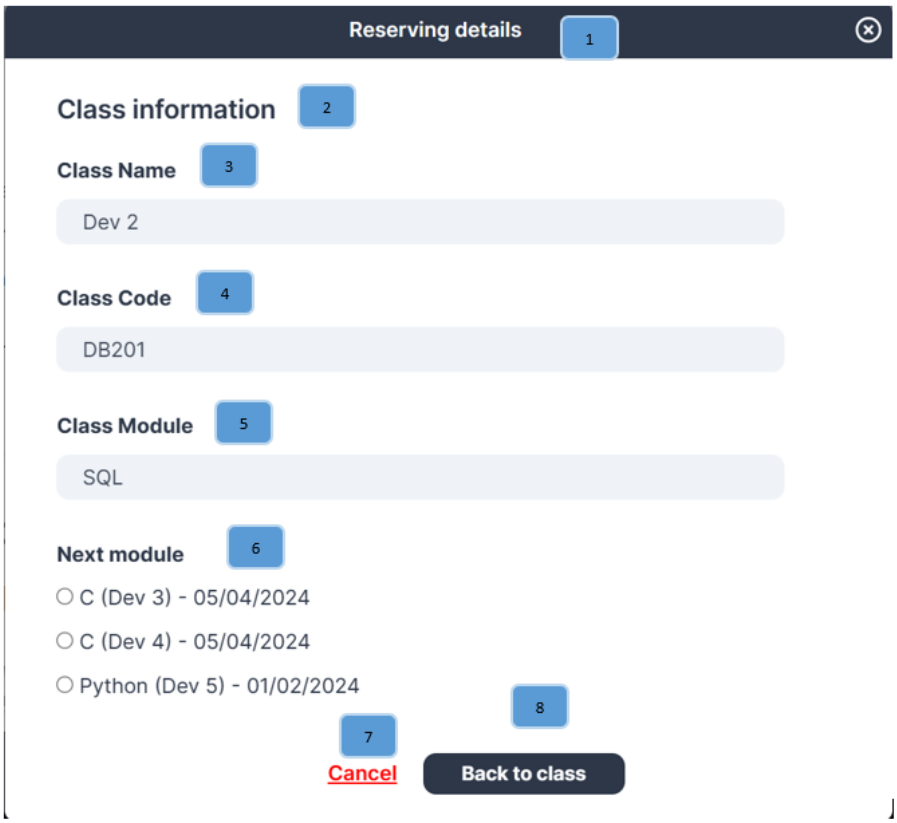


Figure 278 - ReClass Recommendation

Fields:

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Reserving details	Fixed word	Yes	No	Text	String	N/A
2	Class information	Fixed word	Yes	No	Text	String	N/A
3	Class name	Display the Re-Class Name.	Yes	No	TextBox	String	N/A
4	Class code	Display the Re-Class Code.	Yes	No	TextBox	String	N/A
5	Class Modules	Display current modules of the Re Class.	Yes	No	TextBox	String	N/A
6	Next modules	Display the module that students need to return to study	No	Yes	Radio	N/A	N/A

Table 96 - <User Interface-Field> ReClass Recommendation

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Cancel	Click to cancel reclass	N/A	Navigate to the ReClass Dialog Page.
2	Back to class	Click to reclass	N/A	Pop up “Re Class successfully” and Navigate to the ReClass Dialog Page.

Table 97 - <User Interface-Button/Hyperlinks> ReClass Recommendation

5.8. Email Inform/Remind Interface

5.8.1. Email Template List

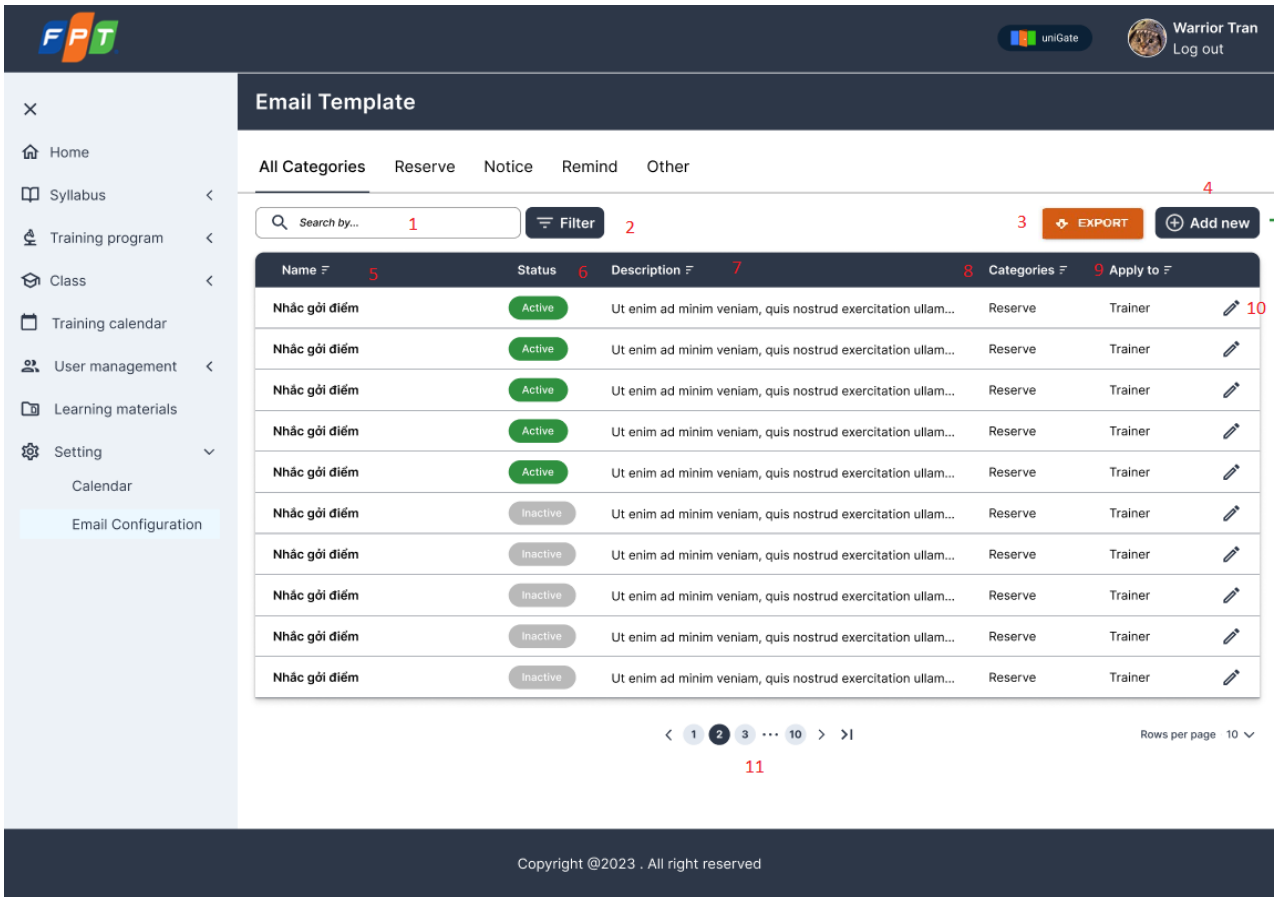


Figure 279 - Email Template List page UI

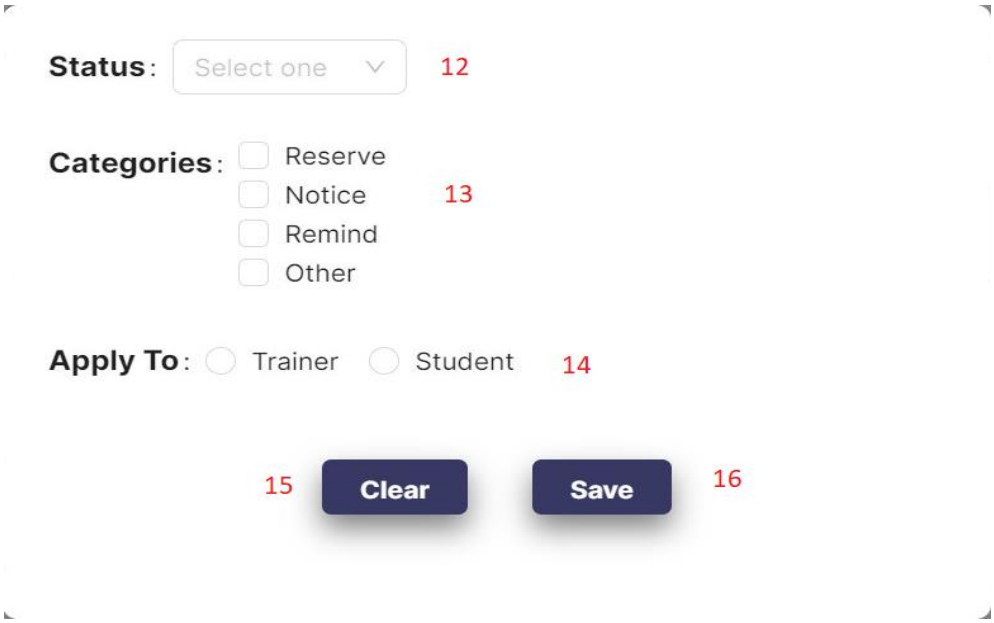


Figure 280 - Search Filter Menu UI

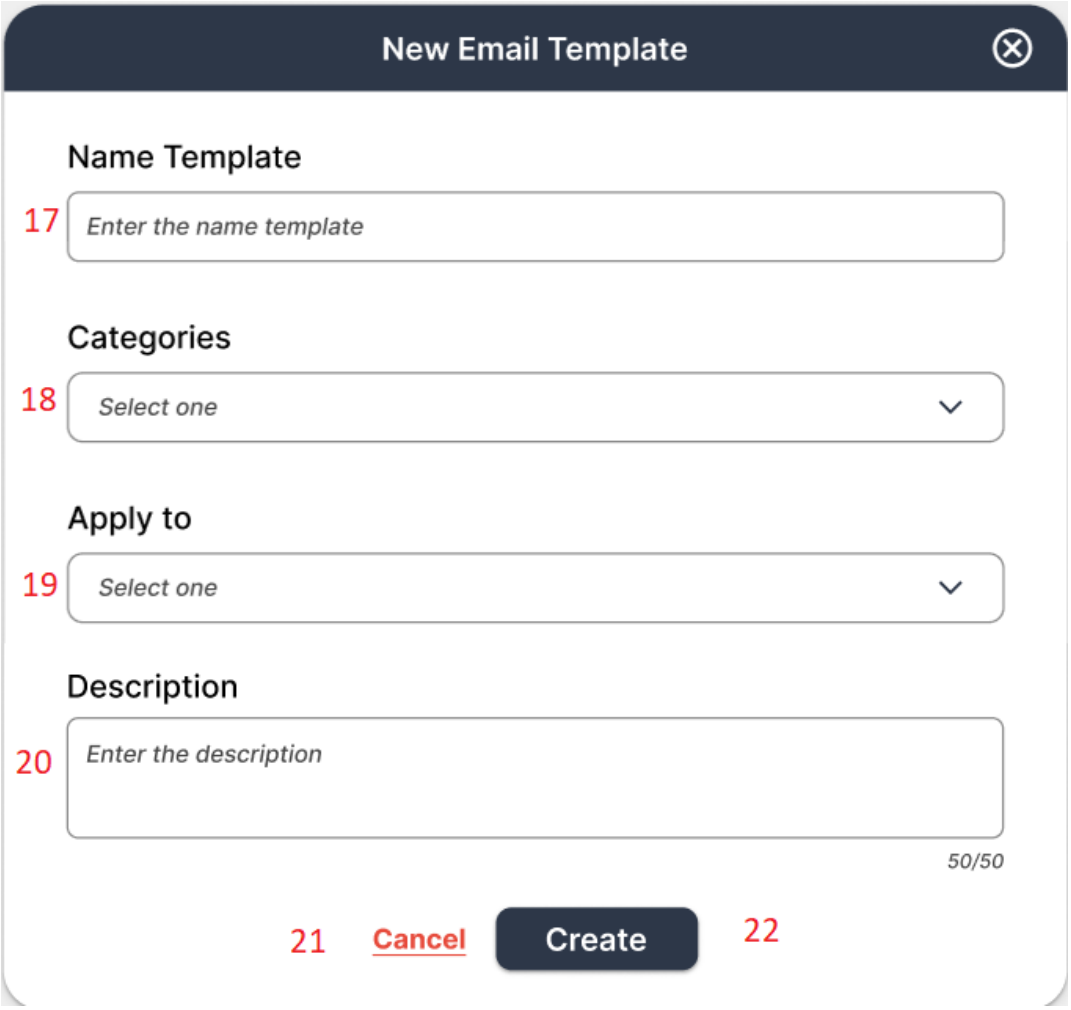


Figure 281 - Add Email Template UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Search by	Search wanted email template	No	No	TextBox	String	N/A
12	Status	Select search status	No	No	Dropdown list	String	N/A
13	Categories	Select search categories	No	No	Checkbox	String	N/A
14	Apply to	Select search apply to	No	No	Radio button	String	N/A
17	Name	Name of email template	No	Yes	TextBox	String	N/A
18	Categories	Categories of email template	No	Yes	Dropdown list	String	N/A
19	Apply to	Who email template send to	No	Yes	Dropdown list	String	N/A

20	Description	Description of email template	No	Yes	TextBox	String	50

Table 98 - <User Interface-Field> Email Template List page

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Filter	Search filter menu	N/A	Open search filter menu
3	Add new	Add new email template	N/A	Open add email template pop up
4	Export	Export data from list to Excel file	N/A	Download excel file
5	Name	Sort email template by name	N/A	List sorted by name
6	Status	Sort email template by status	N/A	List sorted by status
7	Description	Sort email template by description	N/A	List sorted by description
8	Categories	Sort email template by categories	N/A	List sorted by categories
9	Apply to	Sort email template by apply to	N/A	List sorted by apply to
10	Edit	Edit email template	N/A	Navigate to email template detail page
11	Paging (1,2,...)	Move to specific page	N/A	List moved to specific page
15	Clear	Clear all search filter	N/A	Return to email template list page
16	Save	Sort email by search filter	N/A	List sorted by search filter
21	Cancel	Exit add new email template	N/A	Return to email template list page
22	Create	Add email template	N/A	Add email template and return to email template list page

Table 99 - <User Interface-Button/Hyperlinks> Email Template List page

5.8.2. Email template details

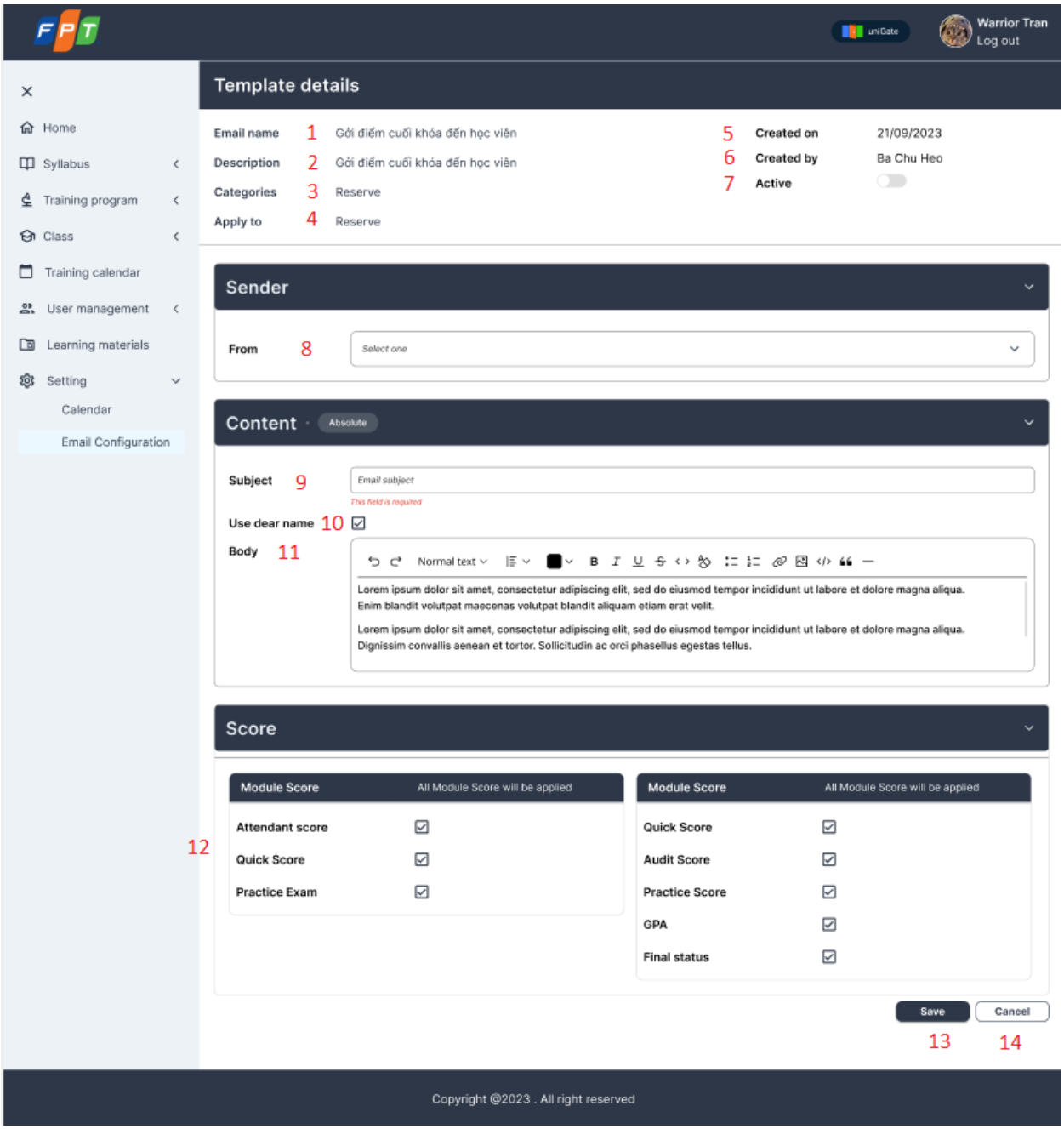


Figure 282 - Email Template Detail UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Email name	Email template name	Yes	Yes	Label	String	N/A
2	Description	Email template description	Yes	Yes	Label	String	N/A

3	Categories	Email template category	Yes	Yes	Label	String	N/A
4	Apply to	Email template apply to	Yes	Yes	Label	String	N/A
5	Create on	Email template create date	Yes	Yes	Label	String	N/A
6	Create by	Email template creator	Yes	Yes	Label	String	N/A
7	Active	Email template status	No	Yes	Toggle switch	String	N/A
8	From	Select email sender	No	Yes	Dropdown list	String	N/A
9	Subject	Email subject	No	Yes	Textbox	String	N/A
10	User dear name	Add "Dear + name" to email content	No	Yes	Checkbox	Boole an	N/A
11	Body	Email body	No	Yes	Rich textbox	String	N/A
12	Module score	Add score table to email content	No	Yes	Checkbox	String	N/A

Table 100 - <User Interface-Field> Email Template Detail page

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
13	Save	Save email details	N/A	Change email template detail and return to email template list page
14	Cancel	Cancel changes	N/A	Return to email template list page

Table 101 - <User Interface-Button/Hyperlinks> Email Template Detail page

5.8.3. Send email to trainer

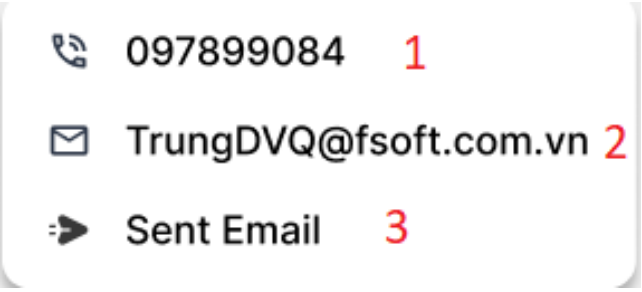


Figure 283 - Trainer info UI

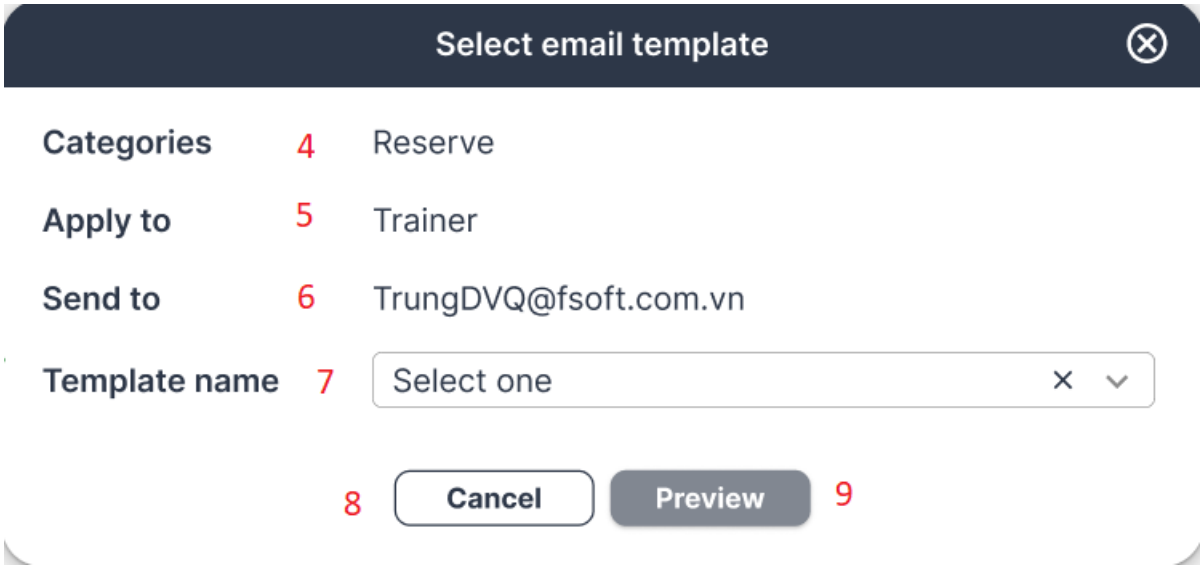


Figure 284 - Send email menu UI

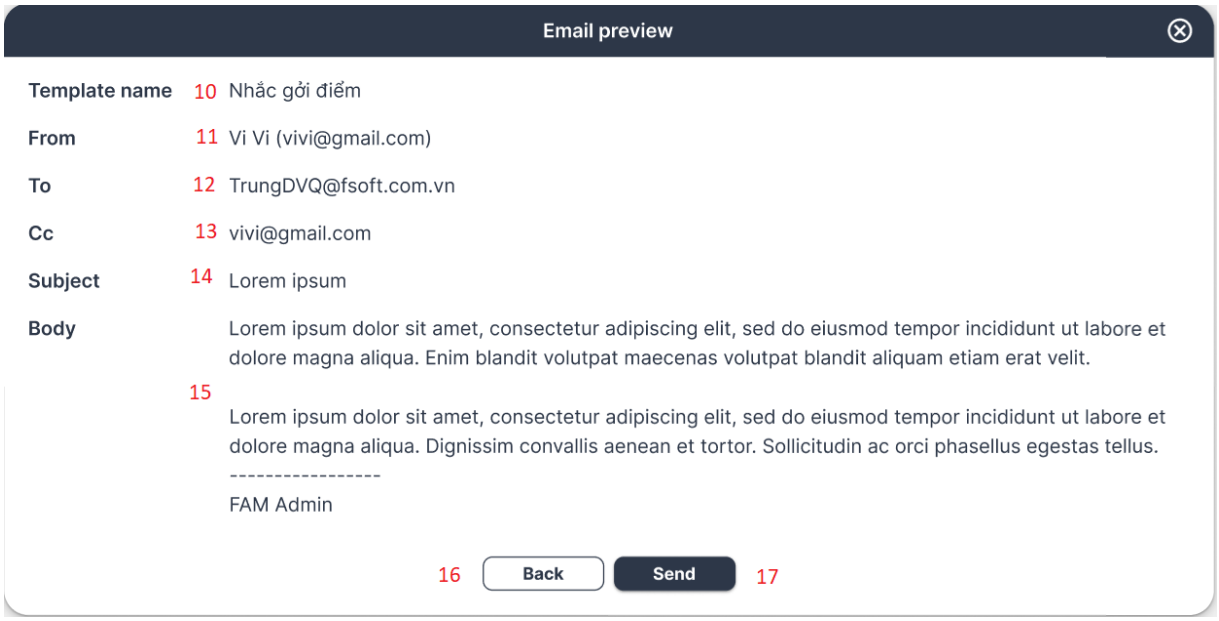


Figure 285 - Send email preview UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Phone number	Trainer's phone number	Yes	Yes	Label	String	N/A
2	Email	Trainer's email	Yes	Yes	Label	String	N/A
4	Categories	Email template category	Yes	Yes	Label	String	N/A
5	Apply	Email template apply	Yes	Yes	Label	String	N/A

	to						
6	Send to	Email receiver	Yes	Yes	Label	String	N/A
7	Templat e name	Select email template	No	Yes	Dropdown list	String	N/A
10	Templat e name	Email template name	Yes	Yes	Label	String	N/A
11	From	Email sender	Yes	Yes	Label	String	N/A
12	To	Email receiver	Yes	Yes	Label	String	N/A
13	Cc	Email cc	Yes	Yes	Label	String	N/A
14	Subject	Email subject	Yes	Yes	Label	String	N/A
15	Body	Email body	Yes	Yes	Label	String	N/A

Table 102 - <User Interface-Field> Send email to trainer

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
3	Send email	Send email to trainer	N/A	Open email template selecting menu pop up
8	Cancel	Cancel sending email	N/A	Return to view class page
9	Preview	Show email preview	N/A	Open email preview
16	Back	Return to email template selecting menu	N/A	Return to email template selecting menu
17	Send	Send email	N/A	Return to view class page

Table 103 - <User Interface-Button/Hyperlinks> Send email to trainer

5.8.4. Send remind reserve email

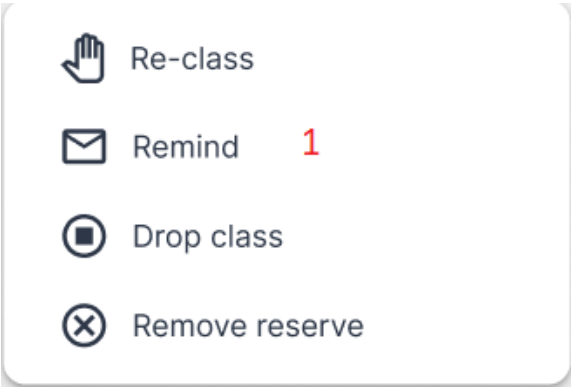


Figure 286 - Reserve action menu UI

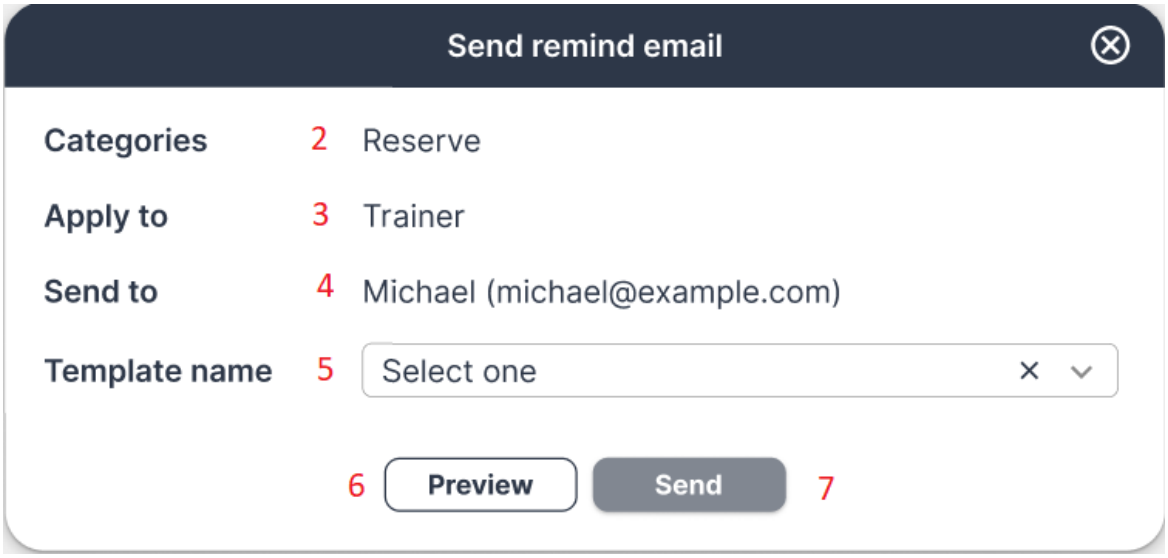


Figure 287 - Send remind email UI

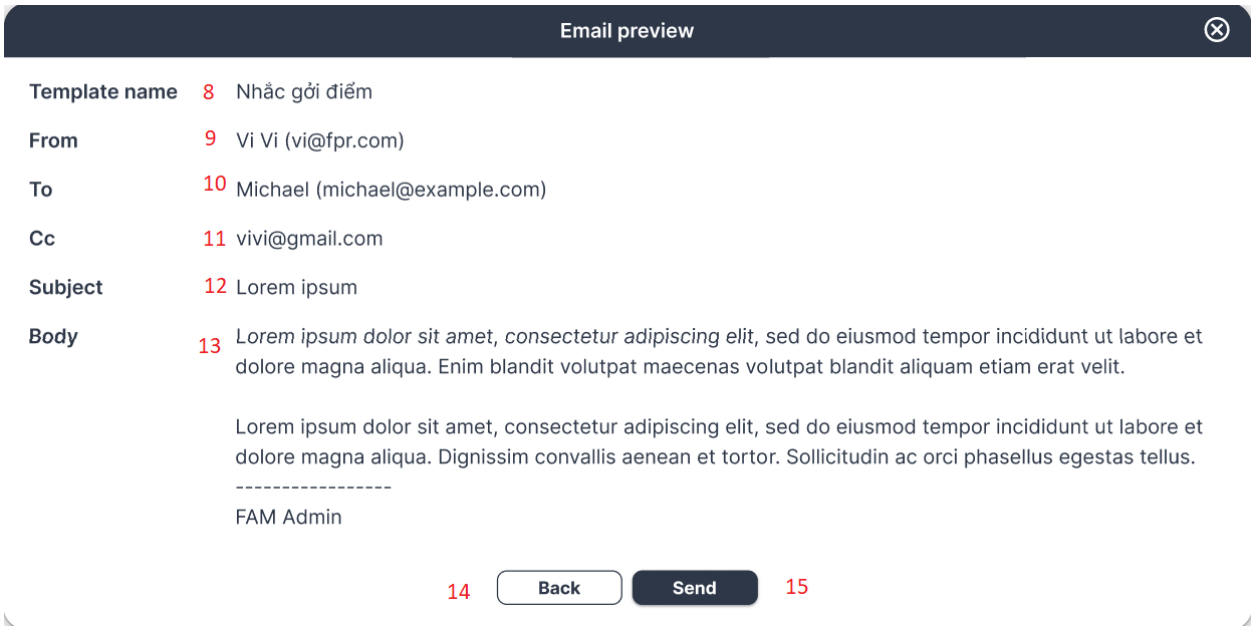


Figure 288 - Send remind email preview UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
2	Categories	Email template category	Yes	Yes	Label	String	N/A
3	Apply to	Email template apply to	Yes	Yes	Label	String	N/A
4	Send to	Email receiver	Yes	Yes	Label	String	N/A
5	Template name	Select email template	No	Yes	Dropdown list	String	N/A

8	Template name	Email template name	Yes	Yes	Label	String	N/A
9	From	Email sender	Yes	Yes	Label	String	N/A
10	To	Email receiver	Yes	Yes	Label	String	N/A
11	Cc	Email cc	Yes	Yes	Label	String	N/A
12	Subject	Email subject	Yes	Yes	Label	String	N/A
13	Body	Email body	Yes	Yes	Label	String	N/A

Table 104 - <User Interface-Field> Send email to reserve

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
1	Remind	Send remind email to reserve	N/A	Open email template selecting menu pop up
6	Preview	Show email preview	N/A	Open email preview
7	Send	Send email	N/A	Return to view reserve list page
14	Back	Return to email template selecting menu	N/A	Return to email template selecting menu
15	Send	Send email	N/A	Return to view reserve list page

Table 105 - <User Interface-Button/Hyperlinks> Send email to reserve

5.8.5. Send email to student

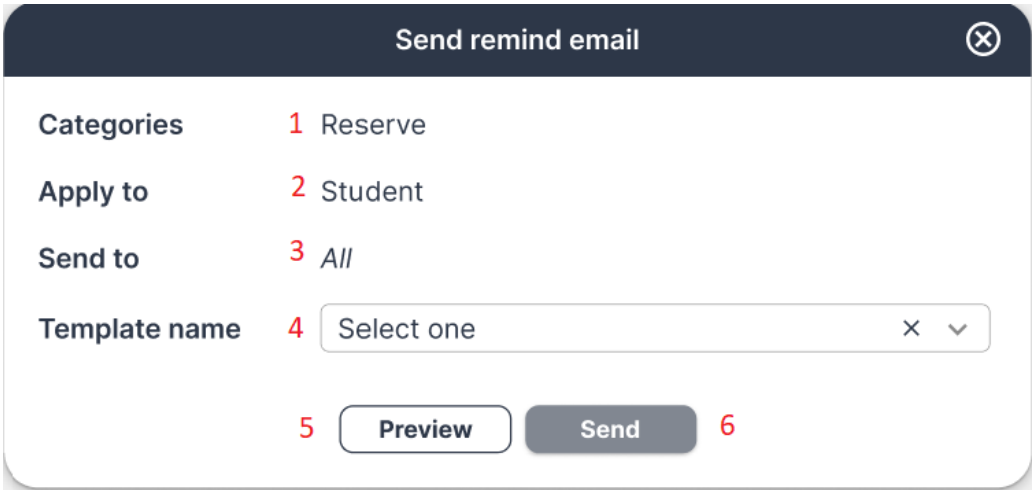


Figure 289 - Send remind email UI

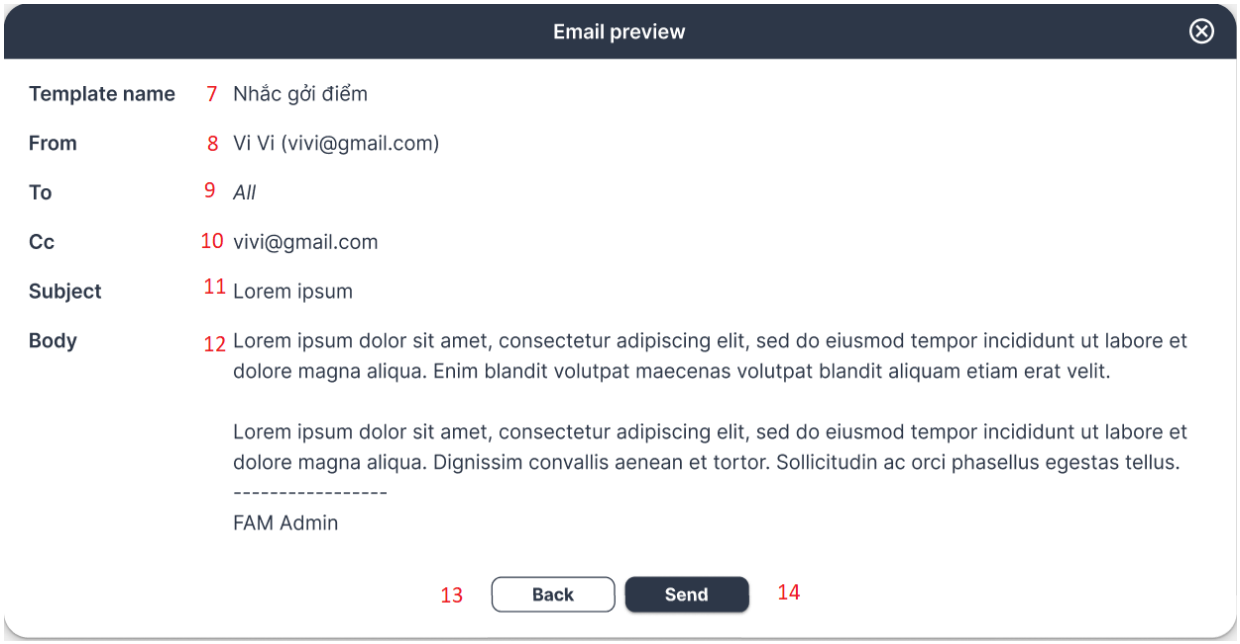


Figure 290 - Send remind email preview UI

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Categories	Email template category	Yes	Yes	Label	String	N/A
2	Apply to	Email template apply to	Yes	Yes	Label	String	N/A
3	Send to	Email receiver	Yes	Yes	Label	String	N/A
4	Template name	Select email template	No	Yes	Dropdown list	String	N/A
7	Template name	Email template name	Yes	Yes	Label	String	N/A
8	From	Email sender	Yes	Yes	Label	String	N/A
9	To	Email receiver	Yes	Yes	Label	String	N/A
10	Cc	Email cc	Yes	Yes	Label	String	N/A
11	Subject	Email subject	Yes	Yes	Label	String	N/A
12	Body	Email body	Yes	Yes	Label	String	N/A

Table 106 - <User Interface-Field> Send email to student

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
5	Preview	Show email preview	N/A	Open email preview

6	Send	Send email	N/A	Return to view reserve list page
13	Back	Return to email template selecting menu	N/A	Return to email template selecting menu
14	Send	Send email	N/A	Return to view reserve list page

Table 107 - <User Interface-Button/Hyperlinks> Send email to student

E. System Test Evidence

- **Goal:** Test all features in the whole FAMS based on core flow.
- **Technique:** xUnit.net

1. User Management Test

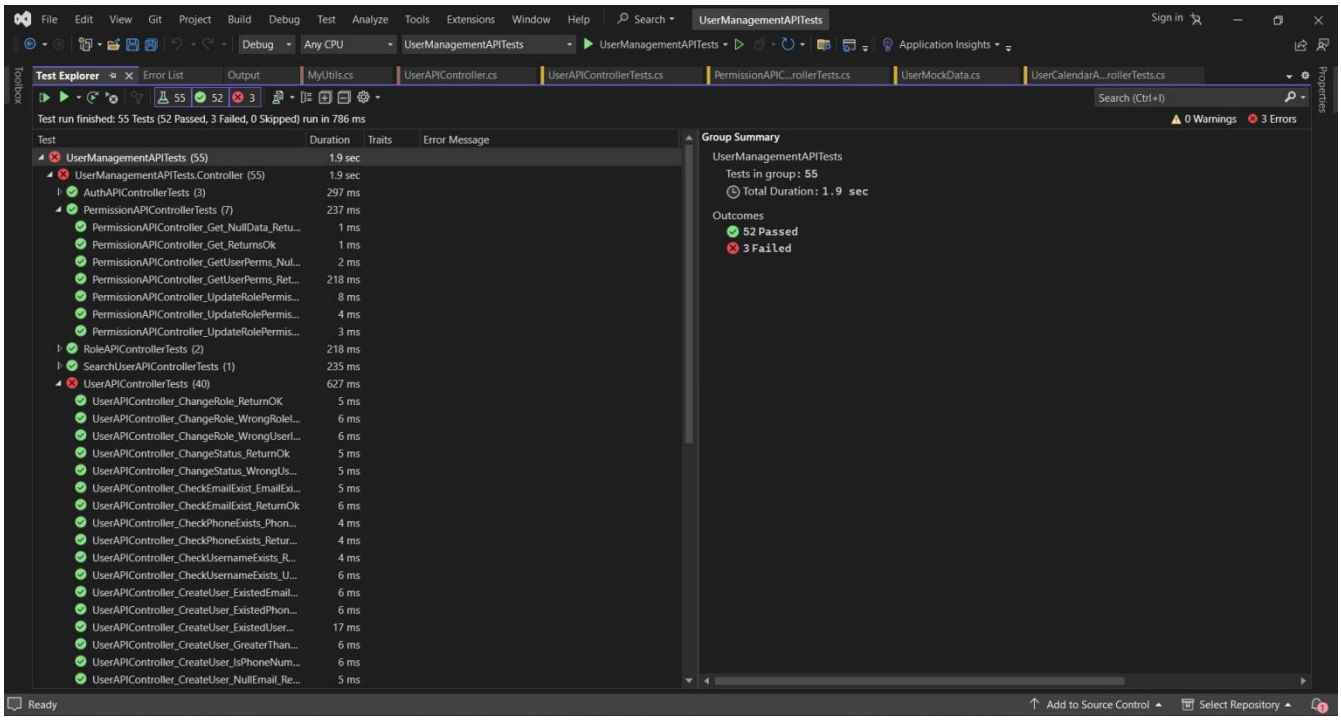


Figure 291 - User Management Test

2. Syllabus Management Test

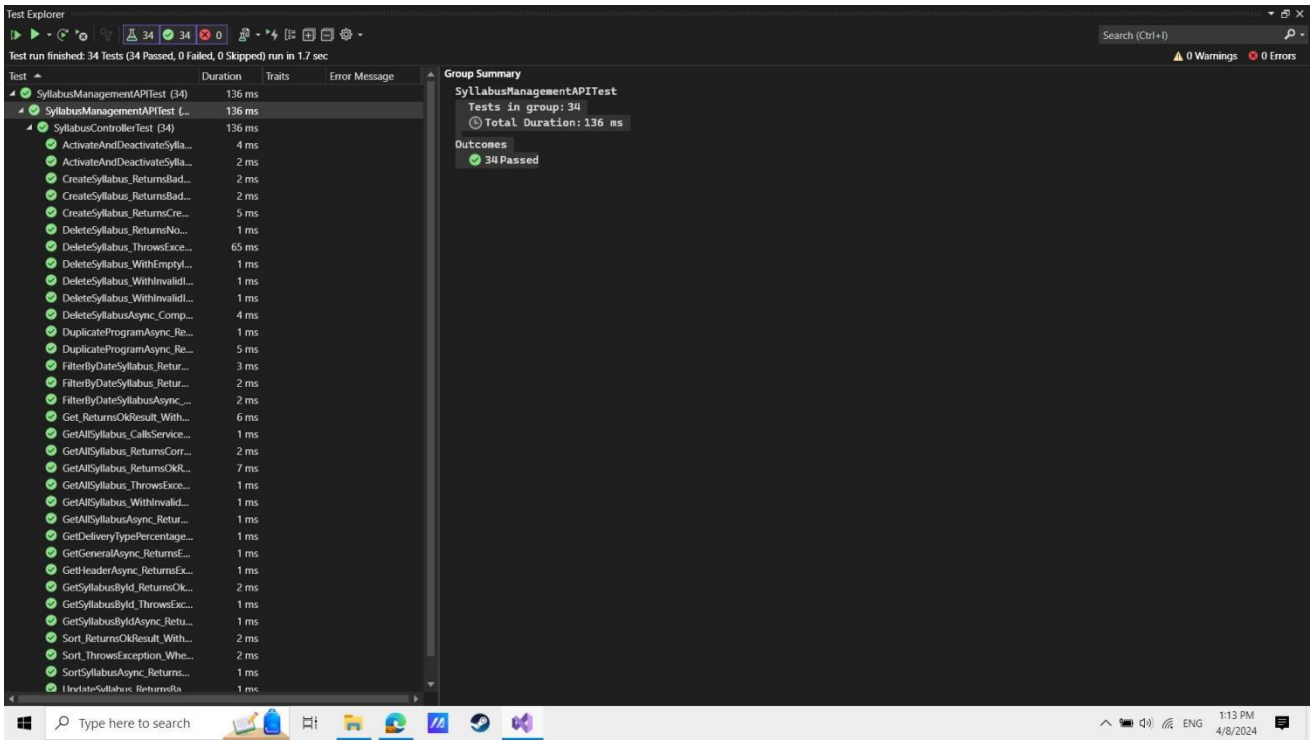


Figure 292 - Syllabus Management Test

3. Training Program Management Test

```
[ApiController]
1 reference
public class TrainingProgramController : ControllerBase
{
    0 references
    public TrainingProgramController(FamsContext context,
        IMapper mapper,
        IOptionsMonitor<AppSettings> monitor,
        IOptionsMonitor<FirebaseCredentials> monitor1,
        IS3Service s3Service,
        IFirebaseService firebaseService);

    [AsyncStateMachine(typeof(<CreateAsync>d__13))]
    [DebuggerStepThrough]
    [HttpPost("api/trainingprograms", Name = "CreateAsync")]
    1 reference
    public Task<IActionResult> CreateAsync([FromBody] CreateTrainingProgramRequest reqObj);
    [AsyncStateMachine(typeof(<DeleteMaterialAsync>d__24))]
    [DebuggerStepThrough]
    [HttpDelete("api/trainingprograms/materials", Name = "DeleteMaterialAsync")]
    1 reference
    public Task<IActionResult> DeleteMaterialAsync([FromQuery] DeleteMaterialRequest reqObj);
    [AsyncStateMachine(typeof(<DeleteTrainingProgramByCode>d__15))]
    [DebuggerStepThrough]
    [HttpDelete("api/trainingprograms/delete/{code}", Name = "DeleteTrainingProgramByCode")]
    1 reference
    public Task<IActionResult> DeleteTrainingProgramByCode([FromRoute] string code);
    [AsyncStateMachine(typeof(<DownloadMaterialAsync>d__21))]
    [DebuggerStepThrough]
    [HttpGet("api/trainingprograms/{code}/materials", Name = "DownloadMaterialAsync")]
    1 reference
    public Task<IActionResult> DownloadMaterialAsync([FromRoute] Guid code);
    [AsyncStateMachine(typeof(<DownloadSingleMaterialAsync>d__22))]
    [DebuggerStepThrough]
    [HttpGet("api/trainingprograms/materials/{id}", Name = "DownloadSingleMaterialAsync")]
    1 reference
    public Task<IActionResult> DownloadSingleMaterialAsync([FromRoute] string id);
    [AsyncStateMachine(typeof(<DuplicateProgramAsync>d__16))]
    [DebuggerStepThrough]
    [HttpPost("api/trainingprograms/duplicate", Name = "DuplicateProgramAsync")]
```

Figure 293 - Training Program Management Test 1

```
[DebuggerStepThrough]
[HttpGet("api/trainingprograms/filter", Name = "FilterAsync")]
1 reference
public Task<IActionResult> FilterAsync([FromQuery(Name = "Status")] string[]? Status = null,
[FromQuery(Name = "CreatedBy")] string CreatedBy = "",
[FromQuery(Name = "ProgramTimeFrameFrom")] DateTime? ProgramTimeFrameFrom = null,
[FromQuery(Name = "ProgramTimeFrameTo")] DateTime? ProgramTimeFrameTo = null,
[FromQuery(Name = "_page")] int _page = 1,
[FromQuery(Name = "_perPage")] string _perPage = "",
[FromQuery] string? sort = null);
[AsyncStateMachine(typeof())]
[AsyncStateMachine(typeof(<GetAllAsync>d__7))]
[DebuggerStepThrough]
[HttpGet("api/trainingprograms/", Name = "GetAllAsync")]
1 reference
public Task<IActionResult> GetAllAsync([FromQuery] int _page = 1, [FromQuery] string _perPage = "");
[AsyncStateMachine(typeof(<GetAllAuthorsAsync>d__14))]
[DebuggerStepThrough]
[HttpGet("api/trainingprograms/authors", Name = "GetAllAuthorsAsync")]
1 reference
public Task<IActionResult> GetAllAuthorsAsync();
[AsyncStateMachine(typeof(<GetByCodeAsync>d__8))]
[DebuggerStepThrough]
[HttpGet("api/trainingprograms/{code}", Name = "GetByCodeAsync")]
1 reference
public Task<IActionResult> GetByCodeAsync([FromRoute] string code);
[AsyncStateMachine(typeof(<GetProgramSyllabusByCodeAsync>d__19))]
[DebuggerStepThrough]
[HttpGet("api/trainingprograms/syllabus/{code}", Name = "GetProgramSyllabusByCodeAsync")]
1 reference
public Task<IActionResult> GetProgramSyllabusByCodeAsync([FromRoute] string code);
[AsyncStateMachine(typeof(<SearchTrainingProgramAsync>d__10))]
[DebuggerStepThrough]
[HttpGet("api/trainingprograms/search", Name = "SearchTrainingProgramAsync")]
1 reference
public Task<IActionResult> SearchTrainingProgramAsync([FromQuery] string? searchValue = "", [FromQuery] int _page = 1, [FromQuery] string _perPage = "", [FromQuery] string?
[AsyncStateMachine(typeof(<SortingAsync>d__11))]
[DebuggerStepThrough]
[HttpGet("api/trainingprograms/sorting", Name = "SortingAsync")]
1 reference
public Task<IActionResult> SortingAsync([FromQuery] string sort, [FromQuery] int _page = 1, [FromQuery] string _perPage = "");
```

Figure 294 - Training Program Management Test 2

```
1 reference
public Task<IActionResult> UpdateMaterialAsync([FromForm] UpdateMaterialRequest reqObj);
[AsyncStateMachine(typeof(<UpdateProgramAsync>d__18))]
[DebuggerStepThrough]
[HttpPut("api/trainingprograms/update", Name = "UpdateProgramAsync")]
1 reference
public Task<IActionResult> UpdateProgramAsync([FromBody] UpdateTrainingProgramRequest reqObj);
[AsyncStateMachine(typeof(<UpdateStatusAsync>d__9))]
[DebuggerStepThrough]
[HttpPatch("api/trainingprograms/{code}", Name = "UpdateStatusAsync")]
1 reference
public Task<IActionResult> UpdateStatusAsync([FromRoute] string code);
[AsyncStateMachine(typeof(<UploadFile>d__20))]
[DebuggerStepThrough]
[HttpPost("api/trainingprograms/uploadFile", Name = "UploadFile")]
1 reference
public Task<IActionResult> UploadFile([FromForm] UploadFile.UploadFileRequest reqObj);
[AsyncStateMachine(typeof(<UploadMaterialAsync>d__23))]
[DebuggerStepThrough]
[HttpPost("api/trainingprograms/materials", Name = "UploadMaterialAsync")]
1 reference
public Task<IActionResult> UploadMaterialAsync([FromForm] UploadMaterialRequest reqObj);
```

Figure 295 - Training Program Management Test 3

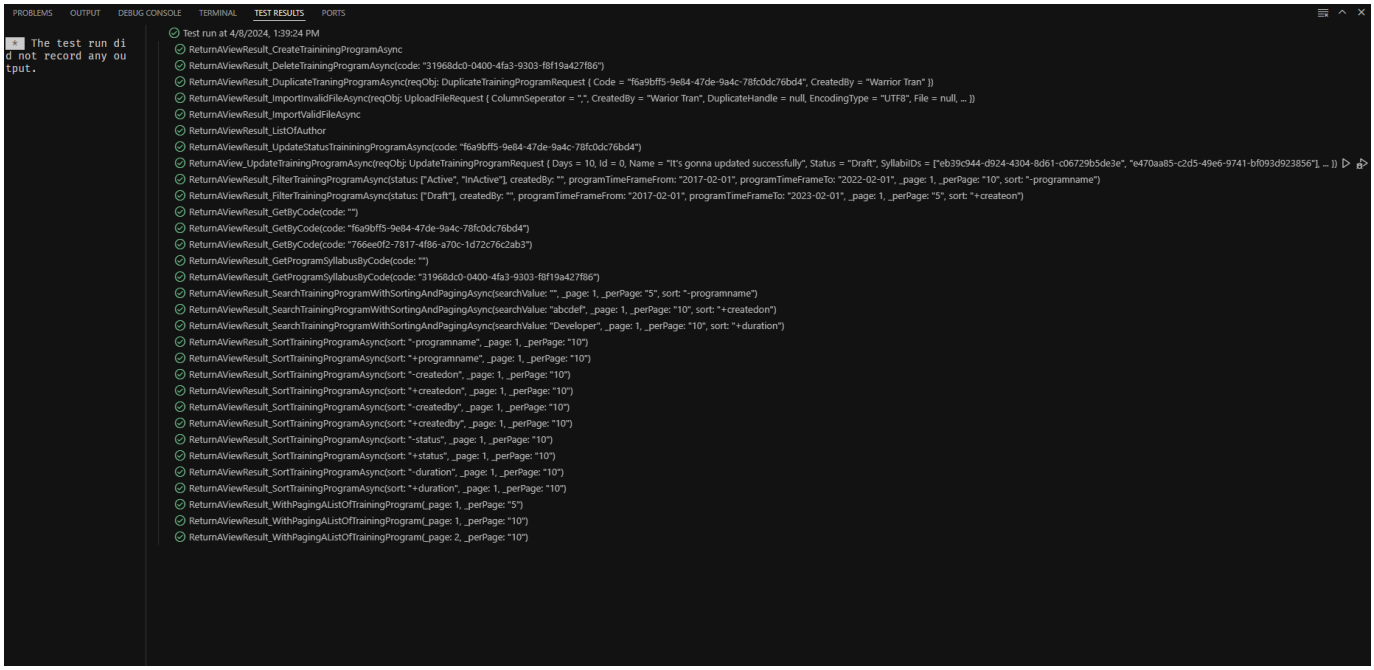


Figure 296 - Training Program Management Test 4

4. Class Management Test

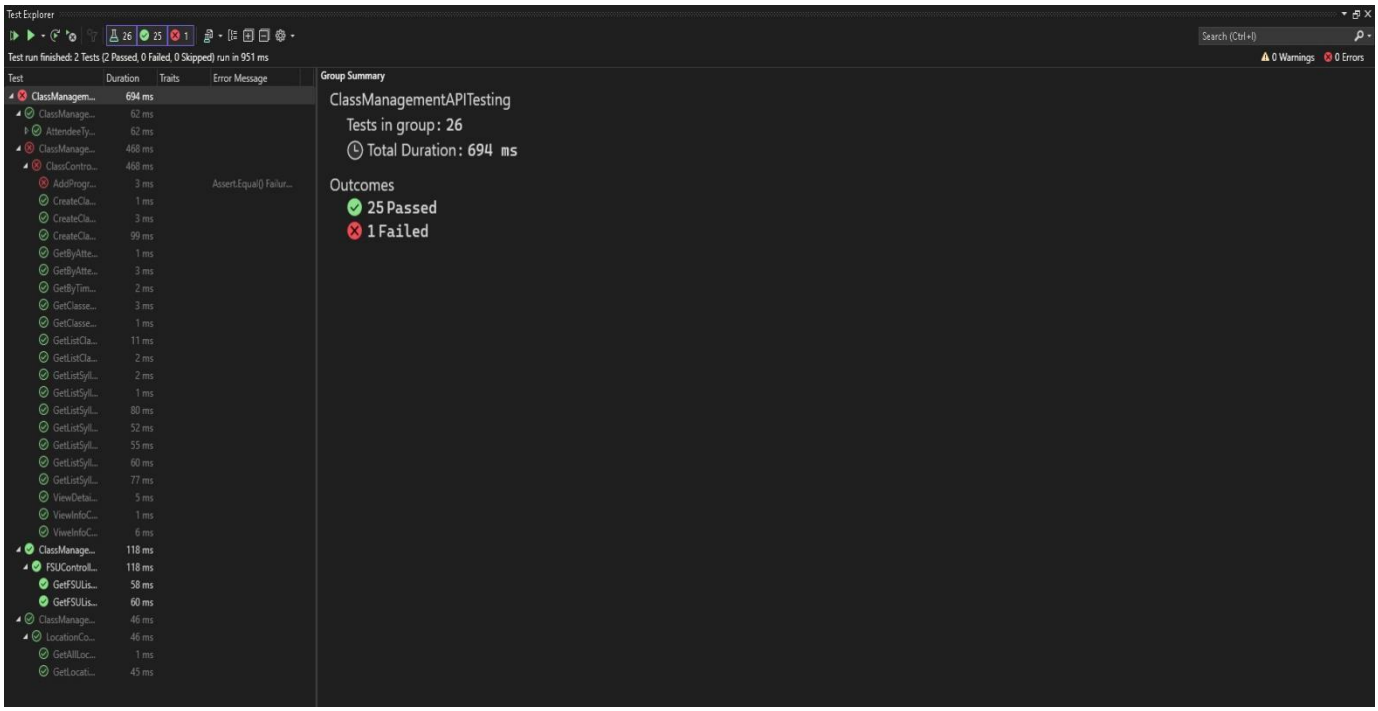


Figure 297 - Class Management Test

5. Student Management Test

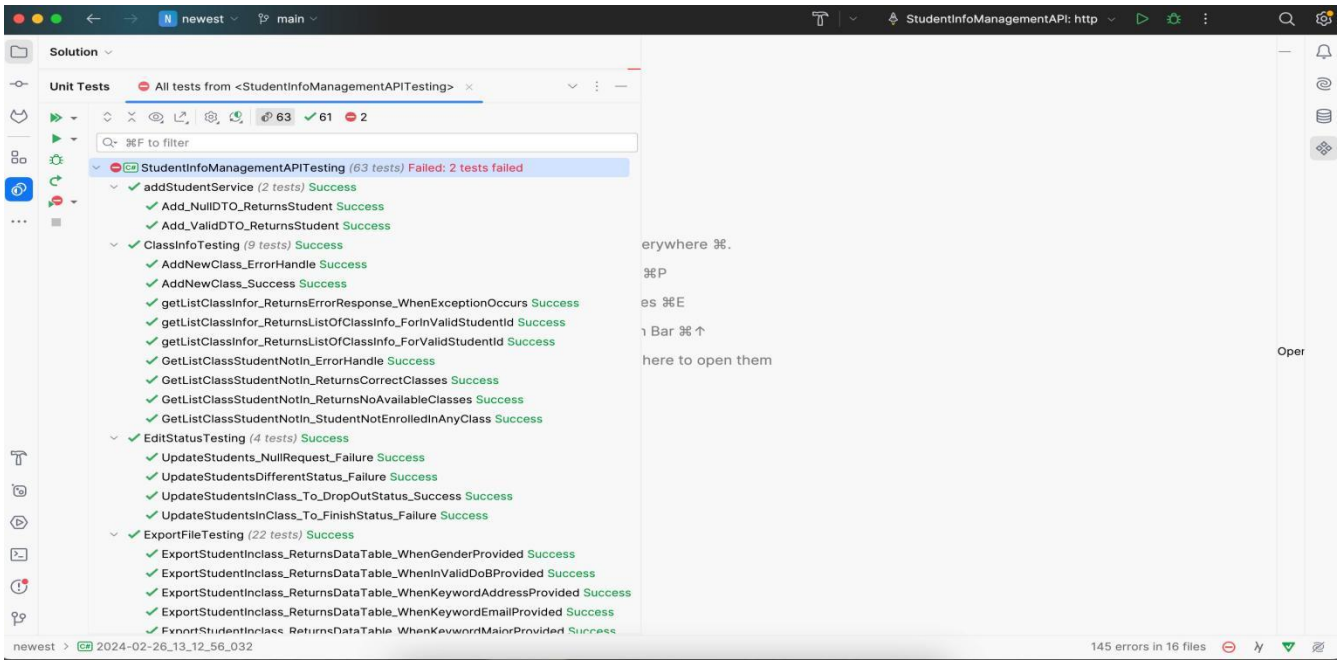


Figure 298 - Student Management Test

6. Score Management Test

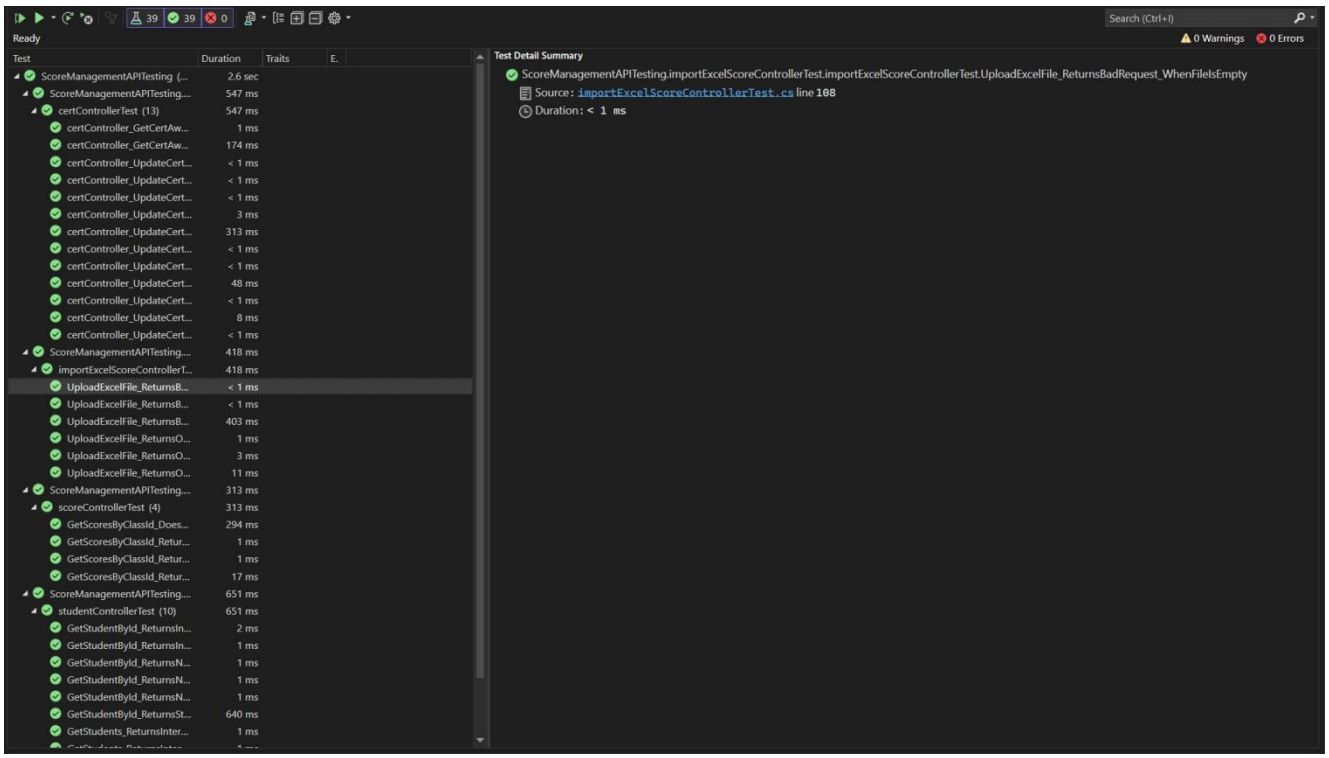


Figure 299 - Score Management Test

7. Reservation Management Test

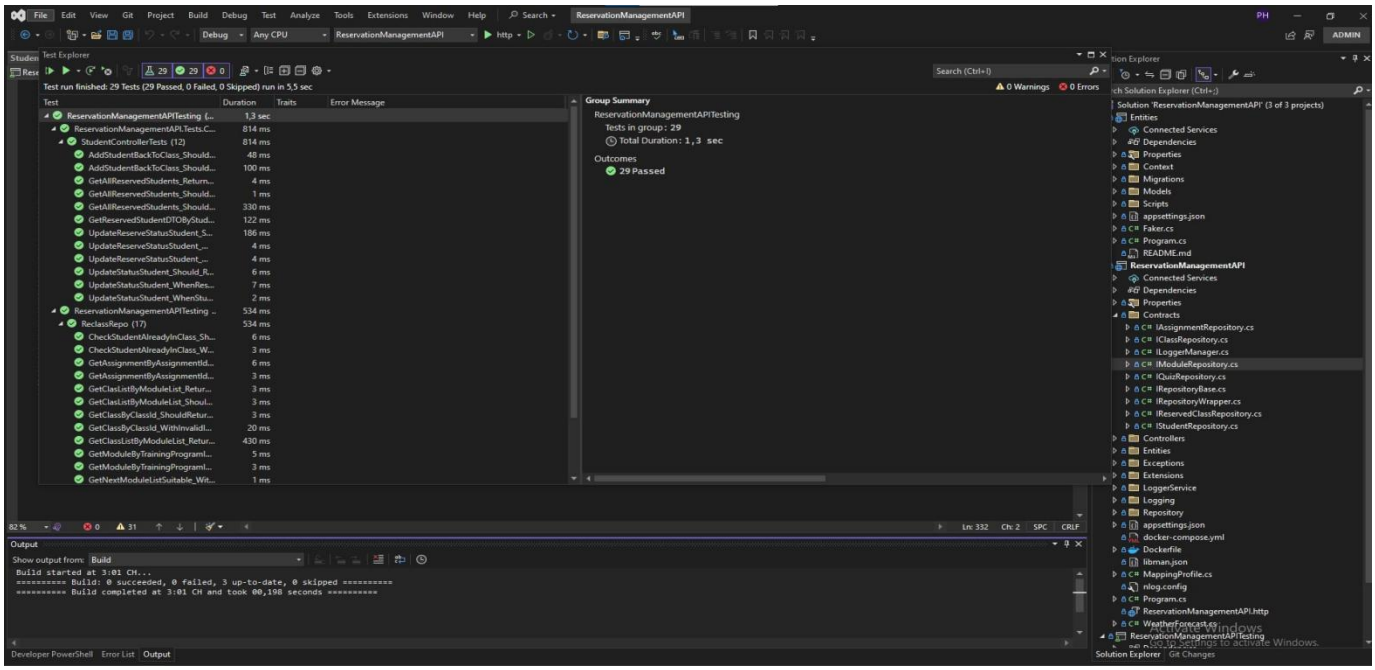


Figure 300 - Reservation Management Test

8. Email Inform/Remind Test

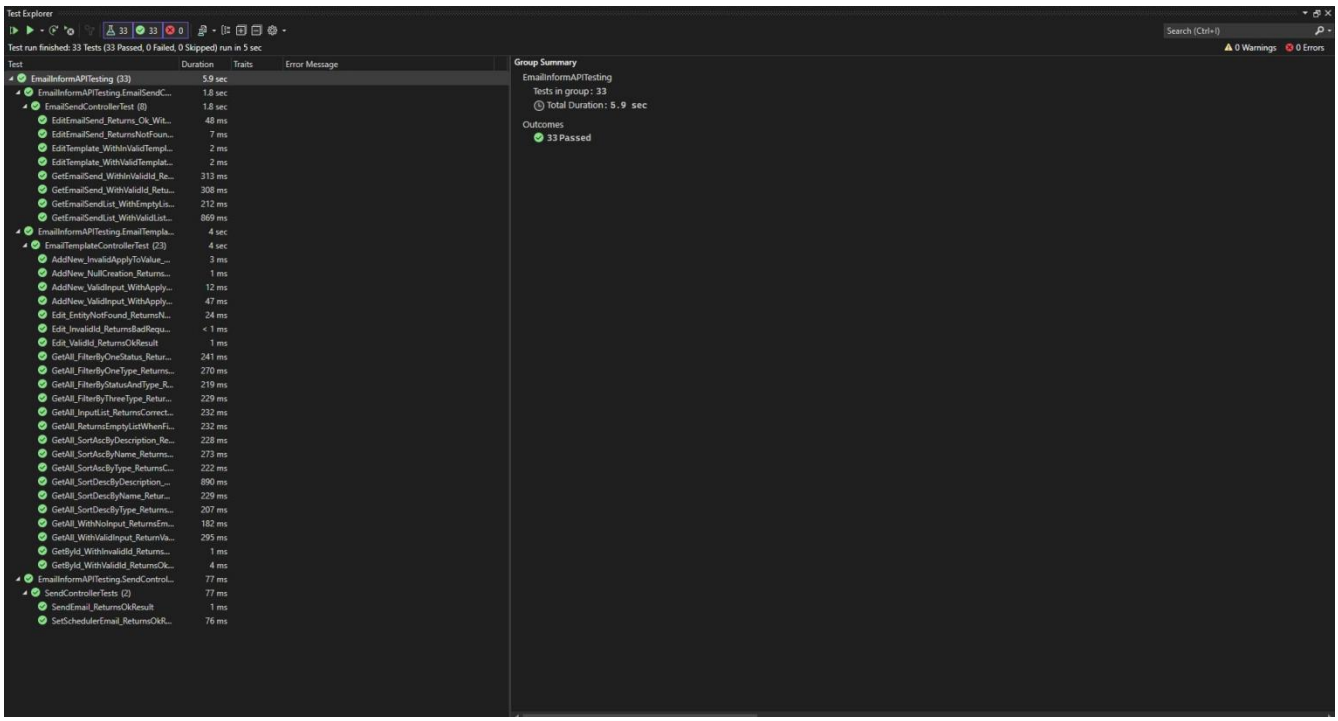


Figure 301 - Email Inform/Remind Test