



TRƯỜNG ĐẠI HỌC FPT

MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document

Online learning platform

SEP490 - G15	
Group Members	Huynh Nguyen Gia Bao - CE171023 Tran Le Hoang Long - CE170879 Mai Nhat Dang - CE170331 Mai Quang Vinh - CE170001 Do Duc Dat - CE171299
Supervisor	Le Thi Phuong Dung
Ext Supervisor	N/A
Capstone Project code	SEP490

- Can Tho, May 2025 -

Table of Contents

Acknowledgement	24
Definition and Acronyms	25
I. Project Introduction	27
1. Overview	27
1.1 Project Information.....	27
1.2 Project Team	27
2. Product Background.....	27
2.1 NodeJs.....	28
2.2 Express.Js	28
2.3 TypeScript	29
2.4 NextJs	30
2.5 Tailwind CSS	30
2.6 MongoDB	31
3. Existing Systems	32
3.1. Quizlet	32
3.2. Knowt	33
3.3. Mochi (Mochi.card).....	34
4. Business Opportunity	35
5. Software Product Vision	35
6. Project Scope & Limitations	36
6.1 Major Features.....	36
6.2 Limitations & Exclusions	41
II. Project Management Plan	42
1. Overview	42
1.1 Scope & Estimation	42
1.2 Project Risks	44
2. Management Approach	44
2.1 Project Process.....	45
2.2 Quality Management	46
2.3 Training Plan.....	48
3. Project Deliverables	48
4. Responsibility Assignments.....	49

4.1 Team & Structures	49
4.2 Role and Responsibility	50
5. Project Communications.....	51
5.1 Communication Plan.....	51
5.2 External Interface.....	51
a. FU Contacts	51
6. Configuration Management.....	51
6.1 Document Management.....	51
6.2 Source Code Management.....	53
6.3 Tools & Infrastructures	54
III. Software Requirement Specification	57
1. Overall description.....	57
2. User Requirements	58
2.1 Overview	58
a. Use Case Diagram.....	58
b. System Actors.....	63
c. Use Case List.....	64
2.2 User Authentication.....	68
2.2.1 Sign in.....	68
2.2.2 Sign up.....	69
2.3 Profile Management	70
2.3.1 View account profile	70
2.3.2 Update account information.....	71
2.3.3 Update password	72
2.3.4 Reset password	73
2.3.5 Delete account	75
2.4 Question Set Management	76
2.4.1 View question sets	76
2.4.2 View question set detail.....	77
2.4.3 Create question sets	78
2.4.4. Update question sets	79
2.4.5 Delete question sets	80
2.4.6 Share question sets.....	81
2.4.7 Export question sets	82
2.4.8 Rate question sets.....	83
2.4.9 Toggle favourite	84

2.4.10 Update question set accessibilities.....	85
2.5 Group Management.....	86
2.5.1 View groups	86
2.5.2 View group detail.....	87
2.5.3 Create groups.....	88
2.5.4 Update groups.....	88
2.5.5 Delete groups.....	89
2.5.6 Send invitations.....	90
2.5.7 Join groups	91
2.5.8 Leave groups	93
2.5.9 Update group accessibilities	94
2.6 Goal Management	94
2.6.1 View goals	94
2.6.2 View goal detail.....	95
2.6.3 Create goals.....	96
2.6.4 Update goal statuses.....	97
2.6.5 Delete goals.....	98
2.7 Settings Management.....	99
2.7.1 View settings	99
2.7.2 Update system appearance	99
2.7.3 Set rest reminder	101
2.7.4 Reset settings	102
2.8 Learner Management	102
2.8.1 View learners	102
2.8.2 View learner profile	103
2.8.3 Update learner accessibilities	104
2.9 Report Management.....	105
2.9.1 Create reports	105
2.9.2 View reports.....	106
2.9.3 View report detail	107
2.9.4 Update report statuses	108
2.10 Field Management	109
2.10.1 View fields.....	109
2.10.2 View field detail	109
2.10.3 Create fields	110
2.10.4 Update fields.....	111

2.10.5 Delete fields	112
2.11 Subject Management.....	113
2.11.1 View subjects	113
2.11.2 View subject detail.....	114
2.11.3 Create subjects.....	115
2.11.4 Update subjects	116
2.11.5 Delete subjects.....	117
2.12 Following Management	118
2.12.1 Follow account.....	118
2.12.2 View follow detail	119
2.13 Question Set Progress Management	120
2.13.1 Mark questions	120
2.13.2 Filter questions	121
2.14 Comment Management.....	122
2.14.1 View comments	122
2.14.2 Filter comments.....	123
2.14.3 Create comments.....	123
2.14.4 Mark question comments.....	125
2.14.5 Update comments	126
2.14.6 Delete comments.....	127
2.14.7 Reply comments	128
2.14.8 Upvote comments.....	129
2.15 Learn Mode Interaction	129
2.15.1 Use learn mode.....	129
2.15.2 Use matching mode	130
2.16 Quiz Mode Interaction	131
2.16.1 Use quiz mode.....	131
2.16.2 Revisit quiz attempts.....	132
2.16.3 View quiz statistics.....	133
2.17 Statistics Interaction	134
2.17.1 View statistics	134
2.17.2 Filter statistics	135
2.18 Additional Features.....	135
2.18.1. Use pomodoro timer.....	135
2.18.2. View achievements	137
2.18.2. View quotes	137

3. Functional Requirements	139
3.1 System Functional Overview.....	139
a. Screen Flow	139
b. Screen Details.....	142
c. Screen Authorization.....	146
d. Non-Screen Functions.....	150
e. Entity Relationship Diagram.....	151
3.2 User authentication	153
3.2.1 Sign in.....	153
3.2.2 Sign up.....	154
3.3 Profile management	155
3.3.1 View account profile	155
3.3.2 Update account information.....	156
3.3.3 Update password	157
3.3.4 Reset password	158
3.3.5 Delete account	159
3.4 Question set management.....	160
3.4.1 View question sets	160
3.4.2 View question set detail.....	161
3.4.3 Create question set	162
3.4.4 Update question sets	163
3.4.5 Delete question sets	164
3.4.6 Share question sets	165
3.4.7 Export question sets	166
3.4.8 Rate question sets.....	167
3.4.9 Toggle favourite	168
3.4.10 Update question set accessibilities.....	169
3.5 Group management.....	170
3.5.1 View groups	170
3.5.2 View group detail.....	171
3.5.3 Create groups.....	172
3.5.4 Update groups.....	173
3.5.5 Delete groups.....	174
3.5.6 Send invitations.....	175
3.5.7 Join groups	176
3.5.8 Leave groups	177

3.5.9 Update group accessibilities	178
3.6 Goal management.....	179
3.6.1 View goals	179
3.6.2 View goal detail.....	180
3.6.3 Create goals.....	181
3.6.4 Update goal statuses.....	182
3.6.5 Delete goals.....	183
3.7 Setting management.....	184
3.7.1 View settings.....	184
3.7.2 Update system appearance	185
3.7.3 Set rest reminder	186
3.7.4 Reset settings	187
3.8 Learner management.....	188
3.8.1 View learners	188
3.8.2 View learner profile	189
3.8.3 Update learner accessibilities	190
3.9 Report management.....	191
3.9.1 Create reports	191
3.9.2 View reports.....	192
3.9.3 View report detail	193
3.9.4 Update report statuses	194
3.10 Field management	195
3.10.1 View fields.....	195
3.10.2 View field detail	196
3.10.3 Create fields	197
3.10.4 Update fields.....	198
3.10.5 Delete fields	199
3.11 Subject management	200
3.11.1 View subjects	200
3.11.2 View subject detail.....	201
3.11.3 Create subjects.....	202
3.11.4 Update subjects	203
3.11.5 Delete subjects.....	204
3.12 Following management	205
3.12.1 Follow account.....	205
3.12.2 View follow detail	206

3.13 Question set progress management.....	207
3.13.1 Mark questions	207
3.13.2 Filter questions	208
3.14 Comment management	209
3.14.1 View comments	209
3.14.2 Filter comments.....	210
3.14.3 Create comments.....	211
3.14.4 Mark question comments.....	212
3.14.5 Update comments	213
3.14.6 Delete comments.....	214
3.14.7 Reply comments	215
3.14.8 Upvote comments.....	216
3.15 Learn mode Interaction	217
3.15.1 Use learn mode.....	217
3.15.2 Use matching mode	218
3.16 Quiz mode interaction	219
3.16.1 Use quiz mode.....	219
3.16.2 Revisit quiz attempts.....	220
3.16.3 View quiz statistics.....	221
3.17 Statistics interaction	222
3.17.1 View statistics	222
3.17.2 Filter statistics	223
3.18 Additional features	224
3.18.1 Use Pomodoro timer.....	224
3.18.2 View achievements	225
3.18.3 View quote	226
4. Non-Functional Requirements	227
4.1 External Interfaces	227
4.1.1 Software Interfaces.....	227
4.1.2 Hardware Interfaces	227
4.1.3 Communication Interfaces.....	227
4.2 Quality Attributes.....	227
4.2.1 Security	227
4.2.2 Performance.....	228
4.2.3 Reliability and availability	228
4.2.4 User interface.....	228

4.2.5 Service and support	228
5. Requirement Appendix	228
5.1 Business Rules	228
IV. Software Design Description	231
1. System Design	231
1.1 System Architecture.....	231
1.2 Package Diagram.....	233
2. Database Design.....	237
3. Detailed Design	262
3.1 User Authentication.....	262
3.1.1 Sign in.....	262
3.1.1.1 <i>Class diagram</i>	262
3.1.1.1.1 Sign in with credential.....	262
3.1.1.1.2 Sign in with OTP	263
3.1.1.1.3 Sign in with Google	264
3.1.1.2 <i>Sequence diagram</i>	265
3.1.1.2.1 Sign in with credential.....	265
3.1.1.2.2 Sign in with OTP	266
3.1.1.2.3 Sign in with Google	267
3.1.2 Sign up.....	268
3.1.2.1 <i>Class diagram</i>	268
3.1.2.1.1 Sign up with credentials.....	268
3.1.2.1.2 Sign up with Google	269
3.1.2.2 <i>Sequence diagram</i>	270
3.1.2.2.1 Sign up with credential	270
3.1.2.2.2 Sign up with Google	271
3.2 Profile Management	272
3.2.1 View account profile	272
3.2.1.1 <i>Class diagram</i>	272
3.2.1.2 <i>Sequence diagram</i>	274
3.2.2 Update account information.....	276
3.2.2.1 <i>Class diagram</i>	276
3.2.2.2 <i>Sequence diagram</i>	277
3.2.3 Update password	280
3.2.3.1 <i>Class diagram</i>	280
3.2.3.2 <i>Sequence diagram</i>	281

3.2.4 Reset password	283
3.2.4.1 <i>Class diagram</i>	283
3.2.4.2 <i>Sequence diagram</i>	284
3.2.5 Delete account	285
3.2.5.1 <i>Class diagram</i>	285
3.2.5.2 <i>Sequence diagram</i>	286
3.3 Question Set Management	287
3.3.1 View question sets	287
3.3.1.1 <i>Class diagram</i>	287
3.3.1.2 <i>Sequence diagram</i>	288
3.3.2 View question set detail.....	290
3.3.2.1 <i>Class diagram</i>	290
3.3.2.2 <i>Sequence diagram</i>	291
3.3.3 Create question sets	293
3.3.3.1 <i>Class diagram</i>	293
3.3.3.2 <i>Sequence diagrams</i>	294
3.3.3.2.1 Create question set.....	294
3.3.3.2.2 Update “create question set” achievement	295
3.3.3.2.3 Update “neuron” achievement.....	297
3.3.4 Update question sets	298
3.3.4.1 <i>Class diagram</i>	298
3.3.4.2 <i>Sequence diagrams</i>	299
3.3.5 Delete question sets	302
3.3.5.1 <i>Class diagram</i>	302
3.3.5.2 <i>Sequence diagram</i>	303
3.3.6 Share question sets	304
3.3.6.1 <i>Class diagram</i>	304
3.3.6.1 Share question set with URL	304
3.3.6.2 Share question set to group.....	304
3.3.6.2 <i>Sequence diagram</i>	305
3.3.6.2.1 Share question set with URL	305
3.3.6.2.2 Share question set to group.....	306
3.3.7 Export question sets	308
3.3.7.1 <i>Class diagram</i>	308
3.3.7.2 <i>Sequence diagram</i>	309
3.3.8 Rate question sets.....	310

3.3.8.1 <i>Class diagram</i>	310
3.3.8.2 <i>Sequence diagram</i>	311
3.3.9 Toggle favourite	312
3.3.9.1 <i>Class diagram</i>	312
3.3.9.2 <i>Sequence diagram</i>	313
3.3.10 Update question set accessibilities.....	314
3.3.10.1 <i>Class diagram</i>	314
3.3.10.1.1 Block question set	314
3.3.10.1.2 Unblock question set	315
3.3.10.2 <i>Sequence diagram</i>	316
3.3.10.2.1 Block question set	316
3.3.10.2.2 Unblock question set	317
3.4 Group Management.....	318
3.4.1 View groups	318
3.4.1.1 <i>Class diagram</i>	318
3.4.1.2 <i>Sequence diagram</i>	319
3.4.2 View group detail.....	321
3.4.2.1 <i>Class diagram</i>	321
3.4.2.2 <i>Sequence diagram</i>	322
3.4.3 Create groups.....	324
3.4.3.1 <i>Class diagram</i>	324
3.4.3.2 <i>Sequence diagram</i>	325
3.4.4 Update groups.....	326
3.4.4.1 <i>Class diagram</i>	326
3.4.4.2 <i>Sequence diagram</i>	327
3.4.5 Delete groups.....	328
3.4.5.1 <i>Class diagram</i>	328
3.4.5.2 <i>Sequence diagram</i>	329
3.4.6 Send invitations.....	331
3.4.6.1 <i>Class diagram</i>	331
3.4.6.2 <i>Sequence diagram</i>	332
3.4.7 Join groups	333
3.4.7.1 <i>Class diagram</i>	333
3.4.7.2 <i>Sequence diagram</i>	334
3.4.8 Leave groups	335
3.4.8.1 <i>Class diagram</i>	335

3.4.8.2 Sequence diagram.....	336
3.4.9 Update groups accessibilities.....	337
3.4.9.1 Class diagram.....	337
3.4.9.1.1 Block group	337
3.4.9.1.2 Unblock group.....	338
3.4.9.2 Sequence diagram.....	339
3.4.9.2.1 Block group	339
3.4.9.2.2 Unblock group.....	340
3.5 Goal Management	341
3.5.1 View goals	341
3.5.1.1 Class diagram.....	341
3.5.1.2 Sequence diagram.....	342
3.5.2 View goal detail.....	343
3.5.2.1 Class diagram.....	343
3.5.2.2 Sequence diagram.....	344
3.5.3 Create goals.....	345
3.5.3.1 Class diagram.....	345
3.5.3.2 Sequence diagram.....	346
3.5.4 Update goal statuses.....	347
3.5.4.1 Class diagram.....	347
3.5.4.2 Sequence diagram.....	348
3.5.5 Delete goals.....	350
3.5.5.1 Class diagram.....	350
3.5.5.2 Sequence diagram.....	351
3.6 Settings Management	352
3.6.1 View settings	352
3.6.1.1 Class diagram.....	352
3.6.1.2 Sequence diagram.....	353
3.6.2 Update system appearance	354
3.6.2.1 Class diagram.....	354
3.6.2.1.1 Change theme	354
3.6.2.1.2 Change language.....	355
3.6.2.1.3 Toggle component	356
3.6.2.2 Sequence diagram.....	357
3.6.2.2.1 Change theme	357
3.6.2.2.2 Change language.....	358

3.6.2.2.3 Toggle component	359
3.6.3 Set rest reminder	360
3.6.3.1 <i>Class diagram</i>	360
3.6.3.2 <i>Sequence diagram</i>	361
3.6.4 Reset setting	362
3.6.4.1 <i>Class diagram</i>	362
3.6.4.2 <i>Sequence diagram</i>	363
3.7 Learner Management	364
3.7.1 View learners	364
3.7.1.1 <i>Class diagram</i>	364
3.7.1.2 <i>Sequence diagram</i>	365
3.7.2 View learner profile	367
3.7.2.1 <i>Class diagram</i>	367
3.7.2.2 <i>Sequence diagram</i>	368
3.7.3 Update learner accessibilities	370
3.7.3.1 <i>Class diagram</i>	370
3.7.3.1.1 Block learner	370
3.7.3.1.2 Unblock learner.....	371
3.7.3.2 <i>Sequence diagram</i>	372
3.7.3.2.1 Block learner	372
3.7.3.2.2 Unblock learner.....	374
3.8 Report Management.....	376
3.8.1 View reports.....	376
3.8.1.1 <i>Class diagram</i>	376
3.8.1.2 <i>Sequence diagram</i>	377
3.8.2 View report detail	378
3.8.2.1 <i>Class diagram</i>	378
3.8.2.2 <i>Sequence diagram</i>	379
3.8.3 Create report.....	380
3.8.3.1 <i>Class diagram</i>	380
3.8.3.1.1 Report question set.....	380
3.8.3.1.2 Report comment.....	381
3.8.3.2 <i>Sequence diagram</i>	382
3.8.3.2.1 Report question set.....	382
3.8.3.2.2 Report comment.....	383
3.8.4 Update report statuses	384

3.8.4.1 <i>Class diagram</i>	384
3.8.4.2 <i>Sequence diagram</i>	385
3.9 Field Management	387
3.9.1 View fields.....	387
3.9.1.1 <i>Class diagram</i>	387
3.9.1.2 <i>Sequence diagram</i>	388
3.9.2 View field details.....	390
3.9.2.1 <i>Class diagram</i>	390
3.9.2.2 <i>Sequence diagram</i>	391
3.9.3 Create field.....	392
3.9.3.1 <i>Class diagram</i>	392
3.10.3.2 <i>Sequence diagram</i>	393
3.9.4 Update field.....	394
3.9.4.1 <i>Class diagram</i>	394
3.9.4.2 <i>Sequence diagram</i>	395
3.9.5 Delete field.....	397
3.9.5.1 <i>Class diagram</i>	397
3.9.5.2 <i>Sequence diagram</i>	398
3.10 Subject Management.....	399
3.10.1 View subjects	399
3.10.1.1 <i>Class diagram</i>	399
3.10.1.2 <i>Sequence diagram</i>	400
3.10.2 View subject details	402
3.10.2.1 <i>Class diagram</i>	402
3.10.2.2 <i>Sequence diagram</i>	403
3.10.3 Create subject	404
3.10.3.1 <i>Class diagram</i>	404
3.10.3.2 <i>Sequence diagram</i>	405
3.10.4 Update subject.....	406
3.10.4.1 <i>Class diagram</i>	406
3.10.4.2 <i>Sequence diagram</i>	407
3.10.5 Delete subject	409
3.10.5.1 <i>Class diagram</i>	409
3.10.5.2 <i>Sequence diagram</i>	410
3.11 Following Management	411
3.11.1 Follow account.....	411

3.11.1.1 <i>Class diagram</i>	411
3.11.1.1.1 Follow account.....	411
3.11.1.1.2 Unfollow account.....	412
3.11.1.2 <i>Sequence diagram</i>	413
3.11.1.2.1 Follow account.....	413
3.11.1.2.2 Unfollow account.....	414
3.11.2 View follow detail	415
3.11.2.1 <i>Class diagram</i>	415
3.11.2.1.1 View followers.....	415
3.11.2.1.2 View followings	416
3.11.2.2 <i>Sequence diagram</i>	417
3.11.2.2.1 View followers.....	417
3.11.2.2.2 View followings	419
3.12 Question Set Progress Management	421
3.12.1 View progress.....	421
3.12.1.1 <i>Class diagram</i>	421
3.12.1.2 <i>Sequence diagram</i>	422
3.12.2 Filter questions	425
3.12.2.1 <i>Class diagram</i>	425
3.12.2.2 <i>Sequence diagram</i>	426
3.12.3 Mark questions	427
3.12.3.1 <i>Class diagram</i>	427
3.12.3.2 <i>Sequence diagram</i>	428
3.13 Comment Management	429
3.13.1 View comments	429
3.13.1.1 <i>Class diagram</i>	429
3.13.1.1.1 View comments	429
3.13.1.1.2 View reply comments	430
3.13.1.2 <i>Sequence diagram</i>	431
3.13.1.2.1 View comments	431
3.13.1.2.2 View reply comments	432
3.13.2 Create comments.....	433
3.13.2.1 <i>Class diagram</i>	433
3.13.2.2 <i>Sequence diagram</i>	434
3.13.3 Reply comments	435
3.13.3.1 <i>Class diagram</i>	435

3.13.3.2 Sequence diagram.....	436
3.13.4 Delete comments.....	437
3.13.4.1 Class diagram.....	437
3.13.4.2 Sequence diagram.....	438
3.13.5 Upvote comments.....	440
3.13.5.1 Class diagram.....	440
3.13.5.2 Sequence diagram.....	441
3.13.6 Update comment	443
3.13.6.1 Class diagram.....	443
3.13.6.2 Sequence diagram.....	444
3.13.7 Filter comments.....	445
3.13.7.1 Class diagram.....	445
3.13.7.2 Sequence diagram.....	446
3.13.8 Mark comment	447
3.13.8.1 Class diagram.....	447
3.13.8.2 Sequence diagram.....	448
3.13.8.2.1 Mark comment	448
3.13.8.2.2 Update “mark comment” achievement.....	449
3.13.8.2.3 Update “neuron” achievement.....	451
3.14 Learn Mode Interaction	452
3.14.1 Use learn mode.....	452
3.14.1.1 Class diagram.....	452
3.14.1.2 Sequence diagram.....	453
3.14.2 Use practice mode	454
3.14.2.1 Class diagram.....	454
3.14.2.2 Sequence diagram.....	455
3.14.3 Use matching mode	457
3.14.3.1 Class diagram.....	457
3.14.3.2 Sequence diagram.....	458
3.15 Quiz Mode Interaction	459
3.15.1 Use quiz mode.....	459
3.15.1.1 Class diagram.....	459
3.15.1.2 Sequence diagram.....	462
3.15.1.2.2.1 Submit quiz	463
3.15.1.2.2.2 Update “submit quiz” achievement.....	465
3.15.1.2.2.3 Update “neuron” achievement.....	467

3.15.2 Revisit quiz attempts.....	468
3.15.2.1 <i>Class diagram</i>	468
3.15.2.2 <i>Sequence diagram</i>	469
3.15.3 View quiz statistics.....	470
3.15.3.1 <i>Class diagram</i>	470
3.15.3.2 <i>Sequence diagram</i>	471
3.16 Statistics Interaction	472
3.16.1 View statistics	472
3.16.1.1 <i>Class diagram</i>	472
3.16.1.1.1 Get total users.....	472
3.16.1.1.2 Get total question sets.....	473
3.16.1.1.3 Get total groups	474
3.16.1.1.4 Get total reports	475
3.16.1.2 <i>Sequence diagram</i>	476
3.16.1.2.1 Get total users.....	476
3.16.1.2.2 Get total question sets.....	477
3.16.1.2.3 Get total groups	478
3.16.1.2.4 Get total reports	479
3.16.2 Filter statistics	480
3.16.2.1 <i>Class diagram</i>	480
3.16.2.1.1 Filter from last week	480
3.16.2.1.2 Filter from last month.....	481
3.16.2.1.3 Filter from last year.....	482
3.16.2.2 <i>Sequence diagram</i>	483
3.16.2.2.1 Filter from last week	483
3.16.2.2.2 Filter from last month.....	484
3.16.2.2.3 Filter from last year.....	485
3.17 Additional Features.....	486
3.17.1 Use Pomodoro timer.....	486
3.17.1.1 <i>Sequence diagram</i>	486
3.17.2 View achievements	487
3.17.2.1 <i>Class diagram</i>	487
3.17.2.2 <i>Sequence diagram</i>	488
3.17.3 View quote	489
3.17.3.1 <i>Sequence diagram</i>	489
4. Class Specification.....	490

4.1 Clerk Classes	490
4.2 Route Classes	496
4.3 Models	498
4.4 Interfaces	502
4.5 Middleware Classes	514
4.6 Controller Classes.....	516
4.7 Service Classes	531
4.8 Repository Classes.....	545
4.9 Error Classes.....	560
V. Software Testing Documentation.....	564
1. Scope of Testing	564
2. Test Strategy	565
2.1 Testing Types.....	565
2.2 Test Levels.....	566
2.3 Supporting Tools	567
3. Test Plan.....	567
3.1 Human Resources	567
3.2 Test Environment.....	568
3.3 Test Milestones	568
4. Test Cases.....	569
5. Test Reports	569
VI. Release Package & User Guides.....	572
1. Deliverable Package	572
2. Installation Guides	572
2.1 System Requirements	572
2.2 Setup Files	572
2.3 Installation Instruction	573
2.3.1 Install Node.js.....	573
2.3.2 Install MongoDB and MongoDB Compass	581
3. User Manual.....	589
3.1 Terms and definitions	589
3.2 System requirements	589
3.3 Application Usage	590
3.3.1. Overview	590
3.3.2. Sign in.....	592
3.3.2.1. Administrator and Learner sign in with credentials	592

3.3.2.2. Administrator and Learner sign in with OTP	595
3.3.2.3. Administrator and Learner sign in with Google.....	600
3.3.3. Sign up.....	603
3.3.3.1. Guest signs up with credential.....	603
3.3.3.2. Guest sign up with Google	606
3.3.4 Sign out	609
3.3.4.1 Learner sign out from header	609
3.3.4.2 Learner sign out from the settings page.....	610
3.3.4.3 Admin sign out from the sidebar	611
3.3.5. View account profile	612
3.3.5.1. Learner view account profile	612
3.3.5.2. Administrator view account profile	614
3.3.6. Update account profile	615
3.3.6.1. Learner change profile image	615
3.3.6.2. Administrator change profile image	617
3.3.6.3. Learner update profile information.....	619
3.3.6.4. Administrator update profile information.....	621
3.3.6.5. Learner update status.....	623
3.3.7 Update password	625
3.3.7.1. Learner change password	625
3.3.7.2. Administrator change password.....	627
3.3.8 Learner delete account	629
3.3.9 Administrator and Learner reset password	631
3.3.10 View question sets	636
3.3.10.1 Learner and Guest view from home page	636
3.3.10.2 Learner view from created / favorited sets	637
3.3.10.3 The Administrator views question sets inside the question sets page.....	639
3.3.11 View question set detail.....	640
3.3.11.1 Learner and guest view question set detail.....	640
3.3.11.2 The Administrators view question sets inside the question sets page.....	642
3.3.12 Learner creates question sets	644
3.3.13 Learner update question sets	648
3.3.13.1 Update question set as a whole.....	648
3.3.13.2 Update individual question	651
3.3.14 Learner delete question sets	653
3.3.15 Share question sets	654

3.3.15.1 Learner and Guest copy Question set URL	654
3.3.15.2 Learner share question set to group.....	656
3.3.15.3 Learner share question set from group	658
3.3.16 Learner export question set.....	660
3.3.17 Learner rate question set.....	662
3.3.18 Learner toggle favourite	663
3.3.19 Learner mark questions	664
3.3.20 Learner and Guest Filter questions.....	665
3.3.20.1 Learner filter marked questions	665
3.3.20.2 Filter with keyword	666
3.3.21 View comments	668
3.3.21.1 Learner and Guest view question set comments	668
3.3.21.2 Learner and Guest view question comments	669
3.3.22 Learner creates comments	670
3.3.22.1 Create question set comment	670
3.3.22.2 Create question comment	671
3.3.23 Learner reply comments.....	673
3.3.23.1 Reply question set comments.....	673
3.3.23.2 Reply question comments	674
3.3.24 Learner update comments.....	676
3.3.24.1 Update question set comments.....	676
3.3.24.2 Update question comments	678
3.3.25 Learner delete comments.....	681
3.3.25.1 Delete question set comments.....	681
3.3.25.2 Delete question comments.....	683
3.3.26 Learner upvote comments.....	686
3.3.26.1 Upvote question set comments.....	686
3.3.26.2 Upvote question comments	687
3.3.27 Mark question comments.....	689
3.3.28 Learner and Guest filter comments	691
3.3.28.1 Filter question set comments	691
3.3.28.2 Filter question comments	692
3.3.29 Learner and Guest use learn mode.....	694
3.3.30 Learner and Guest use matching mode	698
3.3.31 Learner and Guest Use quiz mode	700
3.3.32 Learner view quiz statistics	704

3.3.33 Learner revisit quiz attempt.....	705
3.3.34 View groups	707
3.3.34.1 Learner and Guest view groups when search.....	707
3.3.34.2 Learner view participated groups	710
3.3.34.3 The Administrators view groups inside the groups page.	712
3.3.35 View group detail.....	713
3.3.35.1 Learner and guest view group detail	713
3.3.35.2 The Administrator views group detail.....	714
3.3.36 Learner creates group	715
3.3.37 Learner update groups.....	718
3.3.37.1 Update group settings	718
3.3.37.2 Remove member.....	720
3.3.37.3 Ban member.....	721
3.3.37.4 Change leader	722
3.3.38 Learner deletes groups	723
3.3.39 Learner sends invitations.....	725
3.3.40 Learner joins groups.....	727
3.3.41 Learner leaves groups	728
3.3.42 Learner view goals	729
3.3.42.1 View created goals.....	729
3.3.42.2 View due goals	730
3.3.43 Learner view goal detail.....	731
3.3.44 Learner creates goals	732
3.3.45 Learner update goal statuses.....	734
3.3.45.1 Update goal progress	734
3.3.45.3 Cancel goal	735
3.3.45.3 Finish goal progress	736
3.3.46 Learner deletes goals	737
3.3.47 Learner view settings	738
3.3.48 Change themes	739
3.3.48.1 Administrator change theme	739
3.3.48.2 Learner change theme using header	740
3.3.48.3 Learner change theme inside the Settings page.....	741
3.3.49 Change languages	742
3.3.49.1 Administrator change language.....	742
3.3.49.2 Learner and guest change language using header	743

3.3.49.3 Learner change language inside the Settings page.....	744
3.3.50 Set rest reminder	745
3.3.51 Learner toggle component	746
3.3.51.1 Toggle personal information.....	746
3.3.51.2 Toggle system component.....	748
3.3.52 Learner reset settings	750
3.3.53 View learners	751
3.3.53.1 Learner and Guest view learners from the search result page	751
3.3.53.2 Administrators views learners from the learners page	754
3.3.54 View learner profile	755
3.3.54.1 Learners and guests view learner profile from search result page	755
3.3.54.2 Learners and guests view learner profile from question set detail page	756
3.3.54.3 Administrator views learners from the learners page.....	757
3.3.55 Learner follow learners.....	758
3.3.56 Learner unfollow other learners.....	759
3.3.57 View follow detail	760
3.3.57.1 Learners or guests view learner's followers detail	760
3.3.57.2 Administrator view learner's follow detail	761
3.3.58 Administrator view reports.....	762
3.3.59 Administrator view report detail	763
3.3.60 Learner creates reports.....	764
3.3.60.1 Report question sets.....	764
3.3.60.2 Report comments	765
3.3.61 Administrator update report statuses	766
3.3.62 Administrator update learner visibilities	767
3.3.63 Administrator update question set visibilities	768
3.3.64 Administrator update group visibilities	769
3.3.65 View fields.....	770
3.3.65.1 Learner and guest view fields	770
3.3.65.2 Administrator views fields	771
3.3.66 Administrator view field detail	772
3.3.67 Administrator create fields	773
3.3.68 Administrator update fields	775
3.3.69 Administrator delete fields	777
3.3.70 View subjects	778
3.3.70.1 Learner and guest view subjects.....	778

3.3.70.2 Administrator views subjects.....	779
3.3.71 Administrator view subject detail.....	780
3.3.72 Administrator create subjects.....	781
3.3.73 Administrator Update subjects	783
3.3.74 Administrator delete subjects.....	785
3.3.75 Administrator view statistics	786
3.3.76 Administrator filter statistics	787
3.3.77 Learner and Guest use Pomodoro timer	789
3.3.78 View achievements	793
3.3.78.1 Learner view achievements from achievements page	793
3.3.78.3 Learner and Guest View achievements from learners profile page.	794
3.3.79 View quotes	795
VII. References	796

Acknowledgement

After a long and challenging journey at **the Can Tho campus of FPT University**, we have finally arrived at one of the most significant milestones of our academic lives — the completion of our graduation project. This achievement is far more than the conclusion of a project; it is a testament to the years of perseverance, dedication, and unyielding passion that each of us has poured into our studies. It stands as a reflection of our collective growth — not only academically, but personally and professionally. We are deeply and sincerely thankful to everyone who has supported and encouraged us along this journey. Your presence, guidance, and belief in us have made all the difference.

First and foremost, we extend our heartfelt gratitude to Ms. **Le Thi Phuong Dung**, our dedicated and inspiring mentor. Her unwavering support, patience, and invaluable guidance have been a cornerstone of our success. From the very beginning, her insights and thoughtful advice helped us navigate challenges, correct our course when needed, and push the boundaries of what we thought possible. More than a mentor, she has been a true role model — shaping not just our project, but the way we collaborate, communicate, and envision our future careers. Without her, we simply would not be where we are today.

We would also like to express our profound appreciation to all the **lecturers and faculty members** who have accompanied us throughout our four-year journey. Each class, each lecture, each shared story was a stepping stone that helped us grow. You gave us more than just academic knowledge — you sparked curiosity, encouraged critical thinking, and showed us how theory meets practice in the real world. Your dedication has left a lasting impact on who we are today.

To our **beloved families**, words cannot fully express our gratitude. Your unwavering belief in us, your sacrifices, and your constant emotional support have been our strongest foundation. During our lowest moments, it was your love that lifted us. Every success we celebrate now is just as much yours as it is ours. Your silent strength and endless encouragement have carried us farther than we could have imagined.

We are also sincerely grateful to our **dear friends**, who stood beside us every step of the way. From sharing sleepless nights to celebrating small victories, your companionship has been a source of comfort and motivation. The memories we've created together — the laughter, the challenges, the triumphs — have become a cherished part of this journey, making it not just bearable, but truly unforgettable.

As we look back on the road we've traveled, our hearts are full of appreciation. Every obstacle overcome, every lesson learned, and every hand extended in support has brought us here. With the knowledge and skills we have gained, and with the continued love and encouragement of our mentors, teachers, families, and friends, we feel confident stepping into the next chapter. We carry with us not only dreams, but also the inspiration to create meaningful change, contribute to our community, and continue striving for excellence.

Thank you — from the depths of our hearts — for being a part of this unforgettable journey.

Definition and Acronyms

Acronym	Definition
SEO	Search Engine Optimization
UI	User Interface
SDK	Software Development Kit
SSR	Server-side Rendering
CSR	Client-side Rendering
SSG	Static Site Generation
BR	Business Rule
CLI	Command Line Interface
ERD	Entity Relationship Diagram
PM	Project Manager
SDD	Software Design Description
SRS	Software Requirement Specification
UAT	User Acceptance Test
UC	Use Case
UCD	Use Case Description
API	Application Program Interface
URL	Uniform Resource Locator
OTP	One-time password
VCS	Version Control System
JWT	JSON Web Token
ODM	Object Data Model
SSO	Single Sign-on

SAML	Security Assertion Markup Language
OAuth	Open Authorization

I. Project Introduction

1. Overview

1.1 Project Information

- Project name: Online Learning Platform
- Project code: OLP
- Group name: SEP490-G15
- Software type: Website

1.2 Project Team

Full Name	Role	Email	Mobile
Le Thi Phuong Dung	Supervisor	DungLTP13@fe.edu.vn	0901984504
Huynh Nguyen Gia Bao	Leader	baohngce171023@fpt.edu.vn	0911313132
Tran Le Hoang Long	Member	longtlhce170879@fpt.edu.vn	0848070999
Mai Nhat Dang	Member	dangmnce170331@fpt.edu.vn	0943367449
Mai Quang Vinh	Member	vinhmqce170001@fpt.edu.vn	0819263453
Do Duc Dat	Member	datddce171299@fpt.edu.vn	0812223357

2. Product Background

The way that education is evolving suggests that more all-encompassing solutions are desperately needed. In today's connected and rapidly changing world, educators and students require a single platform that offers robust collaborative study tools, the ability to create flashcards, and a reliable Q&A community. This gap presents an excellent opportunity to innovate and enhance the educational process.

2.1 NodeJs



Figure 2.1: NodeJs¹

Node.js is a powerful, open-source, server-side runtime environment built on Google Chrome's V8 JavaScript engine. It revolutionizes the way developers approach backend development by enabling JavaScript to run outside the browser.

Node.js is renowned for its speed, efficiency, and ability to handle large-scale, I/O-heavy applications such as real-time communication tools, streaming services, and APIs. Its non-blocking, event-driven architecture and unified language for both client and server development simplify complex tasks, reducing development time. Additionally, its vast npm ecosystem and cross-platform compatibility make it an excellent choice for building RESTful APIs, websites, and microservices. Industry leaders like Netflix, LinkedIn, and Uber rely on Node.js for its scalability and performance.

2.2 Express.Js



Figure 2.2: Express.js²

Express.js, often referred to simply as Express, is a minimalist and flexible web application framework for Node.js. It provides a robust set of features for building websites, offering developers a straightforward yet powerful foundation for creating APIs and web servers.

One of Express.js's key strengths lies in its opinionated nature, allowing developers to structure their applications as needed while still benefiting from essential features like middleware support, routing,

¹ https://en.wikipedia.org/wiki/Node.js#/media/File:Node.js_logo.svg

² <https://en.wikipedia.org/wiki/Express.js#/media/File:Expressjs.png>

and HTTP utilities. Its simplicity and modularity make it an ideal choice for both small-scale projects and large, complex enterprise systems.

Express.js also enjoys widespread adoption within the web development community, owing to its seamless integration with Node.js, rich ecosystem of middleware, and active open-source support. Its versatility extends across various use cases, from serving static files to powering dynamic single-page applications or comprehensive RESTful APIs.

In this project, Express.js is used as the backbone for implementing the server-side logic, including routing, data processing, and API endpoint management. Its lightweight design and compatibility with other JavaScript technologies perfectly align with the requirements of the project, ensuring efficient and maintainable backend development.

2.3 TypeScript

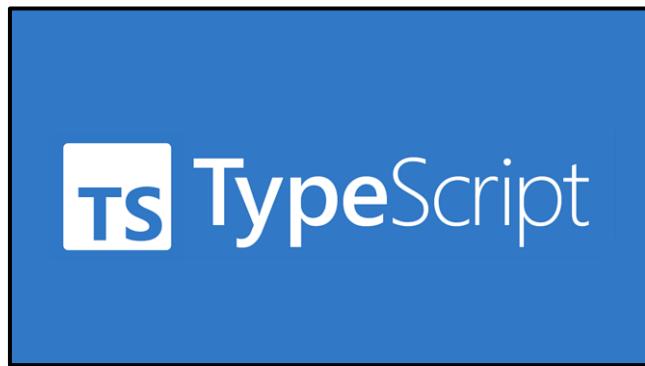


Figure 2.3: TypeScript³

TypeScript is a statically typed superset of JavaScript that enhances code quality and developer productivity by introducing static typing, interfaces, and other advanced features to JavaScript. By enabling developers to define types for variables, function parameters, and object properties, TypeScript allows for early detection of errors during development, leading to more robust and maintainable codebases.

One of the key advantages of TypeScript is its seamless integration with existing JavaScript codebases. Since it is a superset of JavaScript, any valid JavaScript code is also valid TypeScript code. This compatibility allows for gradual adoption, enabling teams to incrementally introduce TypeScript into their projects without significant rewrites. Moreover, TypeScript's support for modern JavaScript features, such as classes and modules, along with its ability to transpile code to older JavaScript versions, ensures compatibility across different environments and browsers.

Incorporating TypeScript into our development workflow can lead to improved code quality, enhanced tooling support, and a more efficient development process. Its growing popularity and strong community support make it a valuable addition to any software engineer's toolkit.

³ <https://blog.haposoft.com/content/images/size/w2000/2022/11/10b88c68-typescript-logo.png>

2.4 NextJs



Figure 2.4: Next.js⁴

Next.js is a React-based framework for building modern web applications, offering a powerful set of features for creating both static and dynamic websites with ease. Developed and maintained by Vercel, Next.js provides an intuitive developer experience by combining server-side rendering (SSR), static site generation (SSG), and client-side rendering (CSR) within a single framework.

A standout feature of Next.js is its file-based routing system, which simplifies the process of creating pages by leveraging the folder structure. Additionally, Next.js enhances performance through automatic code splitting, optimized image handling, and built-in support for incremental static regeneration (ISR), making it an excellent choice for building highly scalable applications.

With built-in support for API routes, seamless integration with modern development tools, and a focus on developer productivity, Next.js is particularly well-suited for creating full-stack applications. It also promotes best practices in web development, such as accessibility, SEO optimization, and performance optimization, ensuring a smooth user experience.

In this project, Next.js serves as the primary framework for building the front-end interface of the platform. Its hybrid rendering capabilities and tight integration with React allow for the creation of responsive, performant, and interactive user experiences, meeting the dynamic requirements of the project.

2.5 Tailwind CSS



Figure 2.5: Tailwind CSS⁵

Tailwind CSS is a modern utility-first CSS framework that simplifies the process of styling web applications by providing a comprehensive set of pre-defined utility classes. Unlike traditional CSS frameworks that rely on pre-styled components, Tailwind enables developers to build custom designs directly within their HTML by combining utility classes.

One of Tailwind's key strengths lies in its flexibility and efficiency. By promoting a "design in your markup" approach, it eliminates the need for writing extensive custom CSS, reducing development

⁴ <https://images.viblo.asia/171ffa2b-6b95-4ec8-838e-f8d8eedbd229.png>

⁵ <https://tailwindcss.com/brand>

time while maintaining full control over the design. Additionally, Tailwind's highly customizable configuration allows developers to extend the framework with custom themes, color palettes, and responsive breakpoints, ensuring adaptability for any project.

Tailwind also excels in delivering consistent and responsive designs. Its utility classes are mobile-first by default, making it easy to create layouts that work seamlessly across different screen sizes. Furthermore, Tailwind's JIT (Just-In-Time) mode enhances performance by generating only the styles used in the project, resulting in smaller CSS bundles and faster load times.

In this project, Tailwind CSS is used to streamline the development of the user interface, enabling rapid prototyping and consistent styling across components. Its focus on utility-first design principles aligns with the project's goals of building a visually appealing and user-friendly web application efficiently.

2.6 MongoDB



Figure 2.6: MongoDB⁶

MongoDB is a popular NoSQL database that provides a flexible and scalable solution for storing and managing data. Unlike traditional relational databases, MongoDB uses a document-oriented data model, where data is stored in JSON-like documents. This approach allows for a more natural representation of hierarchical and complex data structures, making MongoDB an excellent choice for modern applications.

One of MongoDB's key advantages is its schema-less design, which enables developers to store data without defining a fixed schema upfront. This flexibility makes it particularly suitable for applications with evolving requirements. MongoDB also supports powerful querying capabilities, indexing, and aggregation frameworks, ensuring efficient data retrieval and analysis.

In terms of scalability, MongoDB is designed to handle large volumes of data and high-traffic applications through features like horizontal scaling, replication, and sharding. Its distributed architecture ensures high availability and fault tolerance, making it a reliable choice for mission-critical applications.

In this project, MongoDB serves as the primary database for storing and managing data, including user information, application content, and analytics. Its flexibility and seamless integration with Node.js allow for efficient data handling, aligning with the project's requirements for performance, scalability, and ease of development.

⁶ <https://commons.wikimedia.org/wiki/File:Mongodb.png>

3. Existing Systems

3.1. Quizlet

Brief Description: Quizlet is a versatile study platform that helps users master any subject through customizable flashcards, interactive games, and AI-generated practice tests. Designed to support a wide range of learning styles, Quizlet enables students to study more effectively with tools like spaced repetition, smart grading, and collaborative study sets. Whether you're reviewing vocabulary, prepping for exams, or reinforcing classroom learning, Quizlet makes studying faster, more engaging, and tailored to your goals.

Link: <https://quizlet.com/>

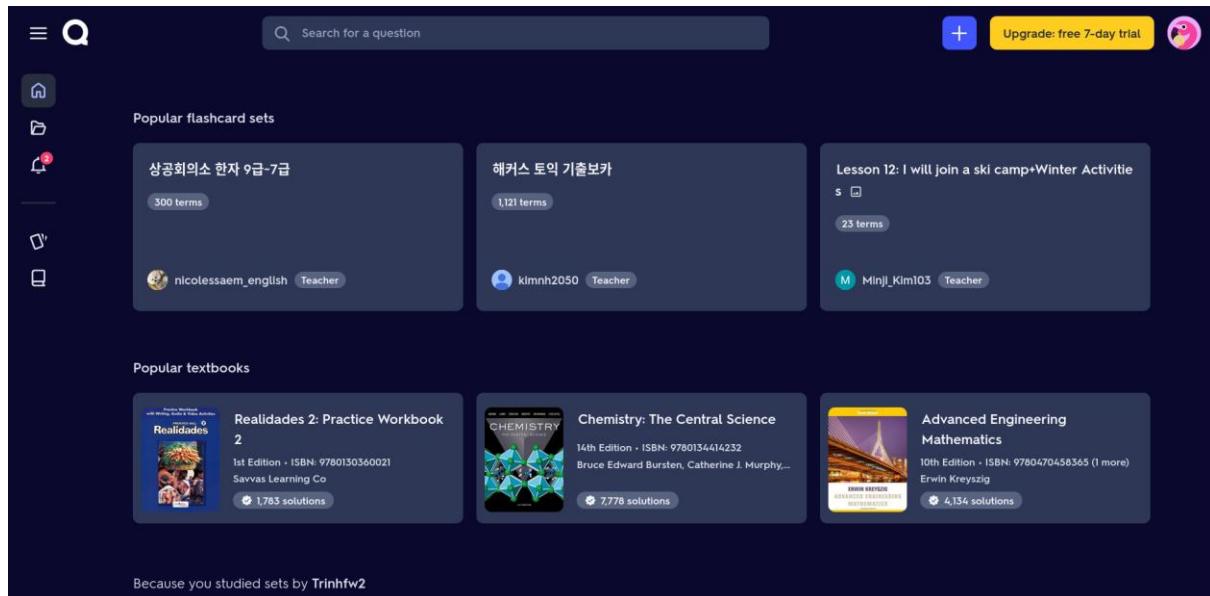


Figure 3.1: Quizlet Homepage

System Actors: Students, teachers, and other educators.

Features:

- Explore and learn any topic through various methods, including flashcards, doing tests and playing games.
- Create and manage your own set of flashcards.
- Create and collaborate with other users using the classroom.
- Tracking user progress for better learning experiences.
- Export and import flashcards with various formats.

Table 1: Summary of the advantages and disadvantages of Quizlet

Pros	Cons
<ul style="list-style-type: none">Easy to use and navigate.Providing a variety of features that help improve learning experience.	<ul style="list-style-type: none">Some features require a paid subscription.Lack of filtering system for flashcard searching or flashcard learning.

- | | |
|---------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> • Large community of users to collaborate with. | <ul style="list-style-type: none"> • May require some technical knowledge for advanced usage. |
|---------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|

3.2. Knowt

Brief Description: Knowt is a free, open-sourced, yet offers an all-in-one learning platform designed to help students study smarter and more efficiently by transforming notes, textbooks, and other learning materials into personalized quizzes and flashcards. With AI-powered tools, automatic quiz generation, spaced repetition, and a sleek, user-friendly interface, Knowt empowers users to retain information faster and perform better on exams. Whether you're organizing your study schedule or preparing for a big test, Knowt offers an intuitive and effective solution that adapts to your unique learning style.

Link: <https://knowt.com/>

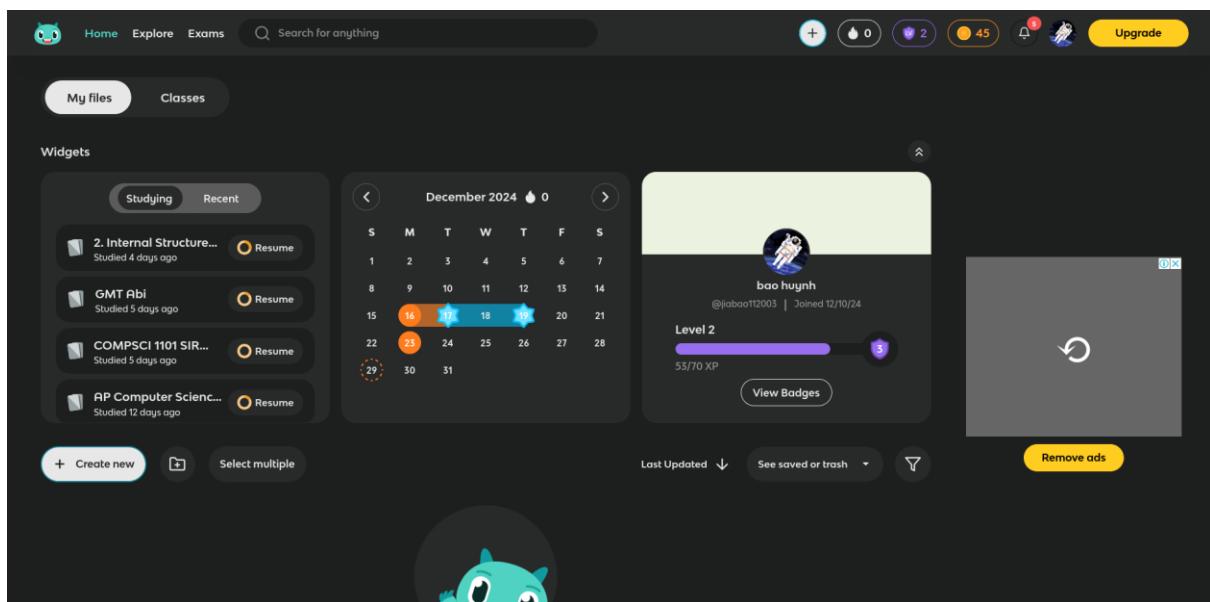


Figure 3.2: Knowt Homepage

System Actors: Students, teachers, and other educators.

Features:

- Explore and learn any topic through various methods, including flashcards, doing tests and playing games.
- Create and manage your own set of flashcards.
- Create and collaborate with other users using the classroom.
- Tracking user progress for better learning experiences.
- Export and import flashcards with various formats.
- Let users create and manage notes that can be shared with other users.
- Provide a system of points and achievements for users.

Table 2: Summary of the advantages and disadvantages of Knowt

Pros	Cons
<ul style="list-style-type: none"> • Free and unlimited usage for most features. • Prioritizes user privacy. • Compatible with various open-source tools and platforms. • Provide a filtering system for flashcards and notes • Support engaging user profile customization with store 	<ul style="list-style-type: none"> • Some features require a paid subscription. • Too much pop-up that can distract users. • May require some technical knowledge for advanced usage. • Only support English language

3.3. Mochi (Mochi.card)

Brief description: Mochi is a minimalist, markdown-based flashcard app that uses spaced repetition to help users retain information more effectively over time. Designed for focused, distraction-free studying, Mochi lets you create linked notes and flashcards in one seamless experience, making it ideal for deep learning and long-term memory. Whether you're learning a new language, studying for exams, or building a knowledge base, Mochi offers a clean, efficient tool that keeps your study sessions smart and structured.

Link: <https://mochi.cards/>

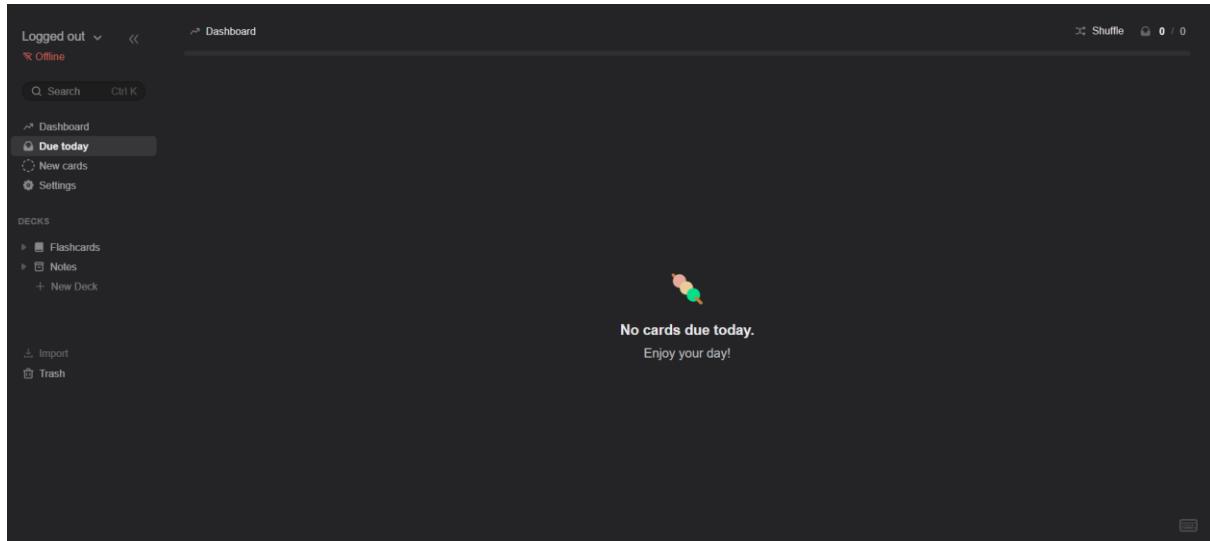


Figure 3.3: Mochi Homepage

System Actors: Students and other learners.

Features:

- Progress tracking monitors your learning journey.
- Flexible creation allows for customization.
- Multiple study modes cater to different learning styles.
- Offline access allows for studying anytime, anywhere.

- Cross-platform compatibility ensures accessibility across devices.

Table 3: Summary of the advantages and disadvantages of Mochi

Pros	Cons
<ul style="list-style-type: none"> • User-friendly interface: Easy to use and highly consistent • Feature-rich: Includes many features as core functionality, eliminating the need for plugins. • Easy tag generation: Supports scripting for automated tag generation. • Affordable: Freemium model with reasonable paid sync costs. 	<ul style="list-style-type: none"> • Limited statistics: The statistics page is quite barebones and provides little information • Lack of note types: May be a limitation for some users. • Export limitations: Cannot export decks with image tags or review history for sharing. • Backup limitations: No option to backup entire collections (including metadata) to a single local file.

4. Business Opportunity

Because of the increased demand for remote learning, customized instruction, and collaborative environments, the educational technology market is growing at a never-before-seen pace. These trends are not merely luxuries; they are essential components of modern education. By directly addressing the shortcomings of current offerings and meeting these significant market demands, our platform aims to become an essential resource for students worldwide. Additionally, this initiative directly supports corporate strategic goals by promoting innovation, ensuring accessibility, and optimizing user engagement in the crucial field of education. Instead of just meeting a need, we believe that by providing a comprehensive and flexible tool, we are empowering education for the future.

The Online Learning Platform is a modern learning platform designed to help students by improving flashcard-based learning. The Online Learning Platform combines flashcard creation, collaborative study tools, and a Q&A community in one place. It focuses on accessibility, personalization, and engagement, meeting the growing demand for remote and personalized learning tools. The Online Learning Platform aims to provide an efficient and enjoyable learning experience while evolving based on user feedback.

5. Software Product Vision

For learners of all ages seeking efficient and engaging methods to master information, The Online Learning Platform is a user-driven, adaptive flashcard platform that personalizes the learning experience based on individual study habits and direct user feedback. Unlike traditional, static flashcard methods, our system leverages user data and community-driven content creation to optimize content organization, track progress, and foster a dynamic learning environment. Users of the Online Learning Platform will experience accelerated learning, improved knowledge retention, and a more enjoyable study process through features directly shaped by their input, ultimately empowering them to achieve their educational and professional goals with greater confidence and

efficiency. This will be achieved through continuous iteration based on user feedback and usage patterns, ensuring our system remains a relevant and effective learning tool adaptable to evolving user needs.

6. Project Scope & Limitations

In this section, we will dive deeper into the scope of this project, clearly defining the objectives, boundaries, and exclusions of our proposed solution. Our team's primary goal is to deliver a comprehensive learning platform designed to help everyone learn effectively by leveraging modern technology, intuitive design, and evidence-based learning methodologies.

Whether users are students preparing for exams, or lifelong learners exploring new subjects, this platform aims to accommodate diverse learning styles and preferences. By integrating intuitive design, advanced features like personalized flashcards, progress tracking, innovative techniques such as gamified learning, spaced repetition, and cross-platform accessibility. Our solution aspires to encourage consistent learning habits and ensure a seamless, engaging learning experience for all users.

However, it is equally important to define the limitations of our solution to manage expectations and maintain focus. This platform will not include advanced features such as live tutoring sessions, or AI-driven adaptive learning in its initial version. Additionally, while the platform supports individual learning, it is not intended to replace formal education systems or provide accredited certifications. By clearly outlining the scope of this project, we aim to ensure that our efforts are concentrated on delivering a high-quality, user-centric product within the defined boundaries.

6.1 Major Features

FE-01: System registration: Allow guests to create an account with their credentials.

FE-02: System authentication: Allow learners and the Administrator to access and manage system accessibility using their credentials.

FE-03: Profile management: Allow learners and the Administrator to view, update their account profile information, learners can follow or unfollow other learner accounts, learners can delete their own account.

FE-04: System exploration: Allow users to browse and interact with system data including question sets, groups created by the community, other learner accounts... Learners can view their own achievements.

FE-05: Question set interaction: Allow learners and guests to view and share question sets, report inappropriate sets, view and report comments created by other learners, learners can add a question set to their favourite list, filter terms based on their preferences and export the question set.

FE-06: Comments management: Allow learners to create and manage their comments inside a question set, learners can also reply, upvote other learners' comments.

FE-07: Question set contribution: Allow learners to rate stars, create and manage comments for terms inside a question set, and the author of the question set can mark the feedback inside the term.

FE-08: Learn mode utilization: Allow learners and guests to learn a question set and customize how they want to learn it, including learning with flashcards and matching games.

FE-09: Quiz mode utilization: Allow learners and guests to take quiz attempts at a question set and customize how they want to do it.

FE-10: Question set tracking: Allow learners to keep track of their progress on a question set or their previous quiz attempts.

FE-11: Question sets management: Allow learners to create and manage their created question sets. The Administrator can block or unblock question sets.

FE-12: Groups management: Allow learners to create and manage their groups. Group members can share question sets, send invitations to other learners, based on group leader permissions. The Administrator can block or unblock groups.

FE-13: Goals management: Allow learners to create and manage their own goals.

FE-14: Pomodoro utilization: Allow learners and guests to utilize Pomodoro timer to manage their own learning session.

FE-15: Setting customization: Allow users to update system settings or appearance based on their references.

FE-16: Learners management: Allow the Administrator to manage learners in the system.

FE-17: Reports management: Allow the Administrator to manage reports sent by learners or guests.

FE-18: Fields management: Allow the Administrator to create and manage question set categories in the system.

FE-19: Subjects management: Allow the Administrator to create and manage question set sub-categories in the system.

FE-20: Statistics interaction: Allow the Administrator to view and filter statistics from the system.

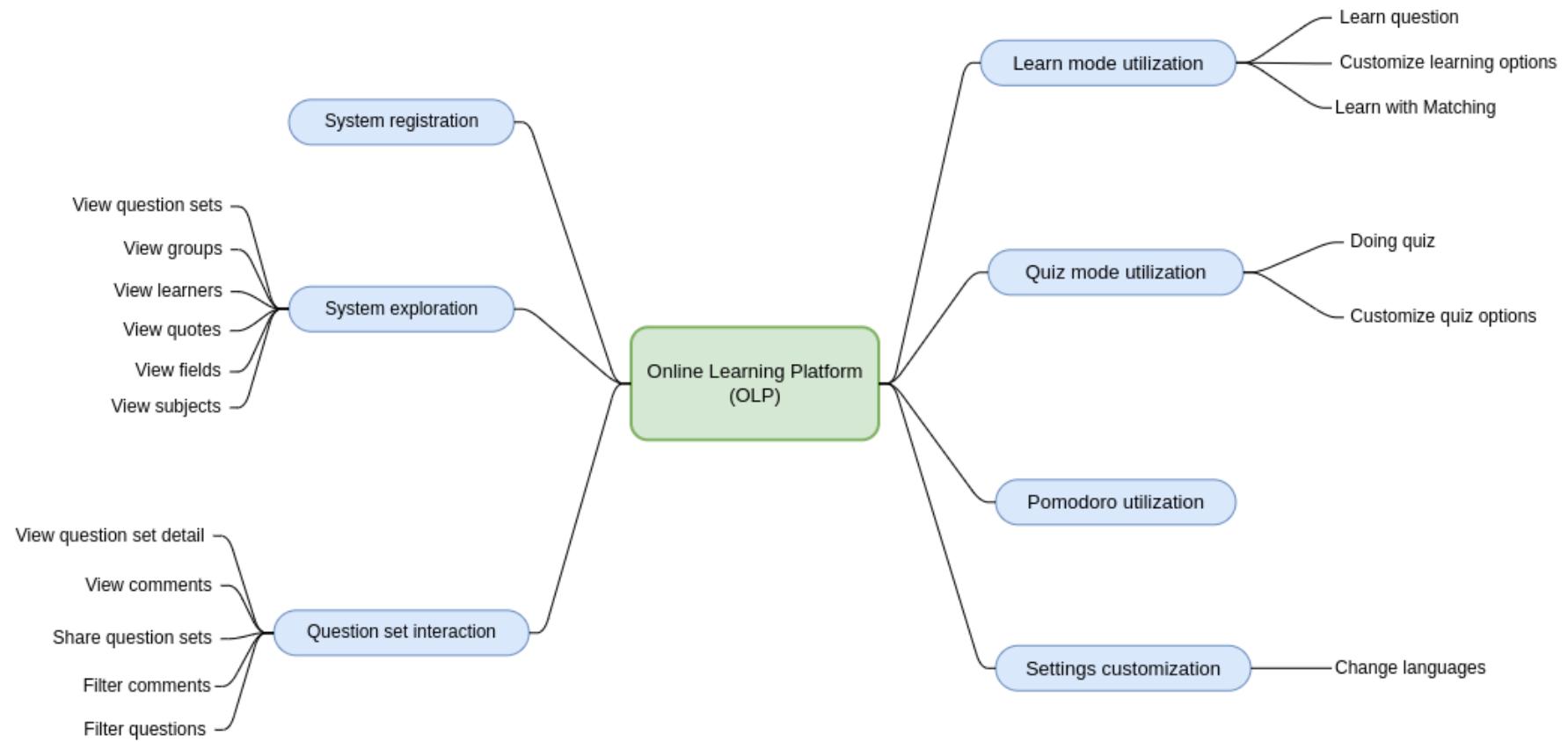


Figure 6.1.1.a: Guest website main feature⁷

⁷ [guest mind map web.png](#)

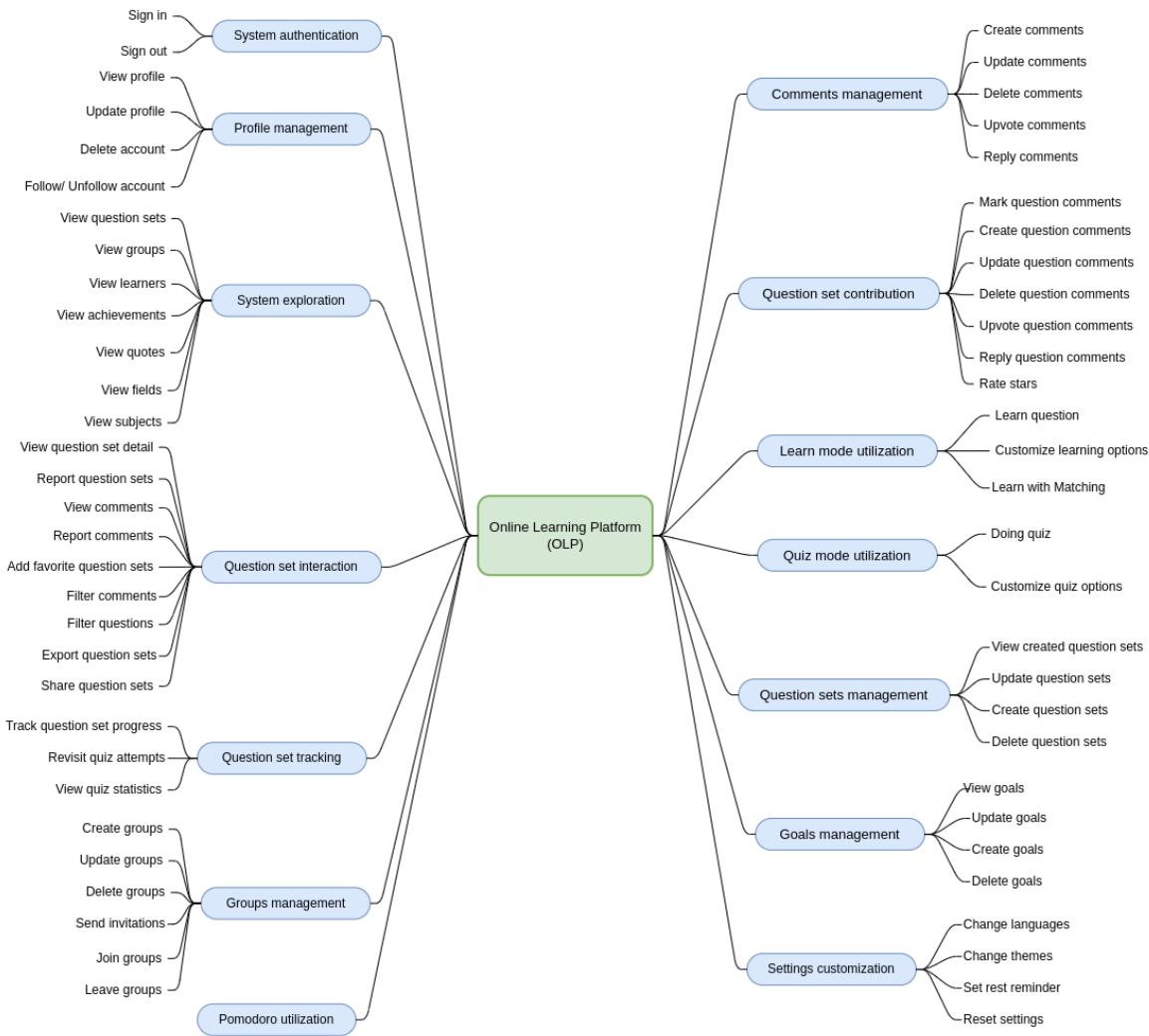


Figure 6.1.2.a: Learner website main feature⁸

⁸ [learner-mind-map-web.drawio.png](#)

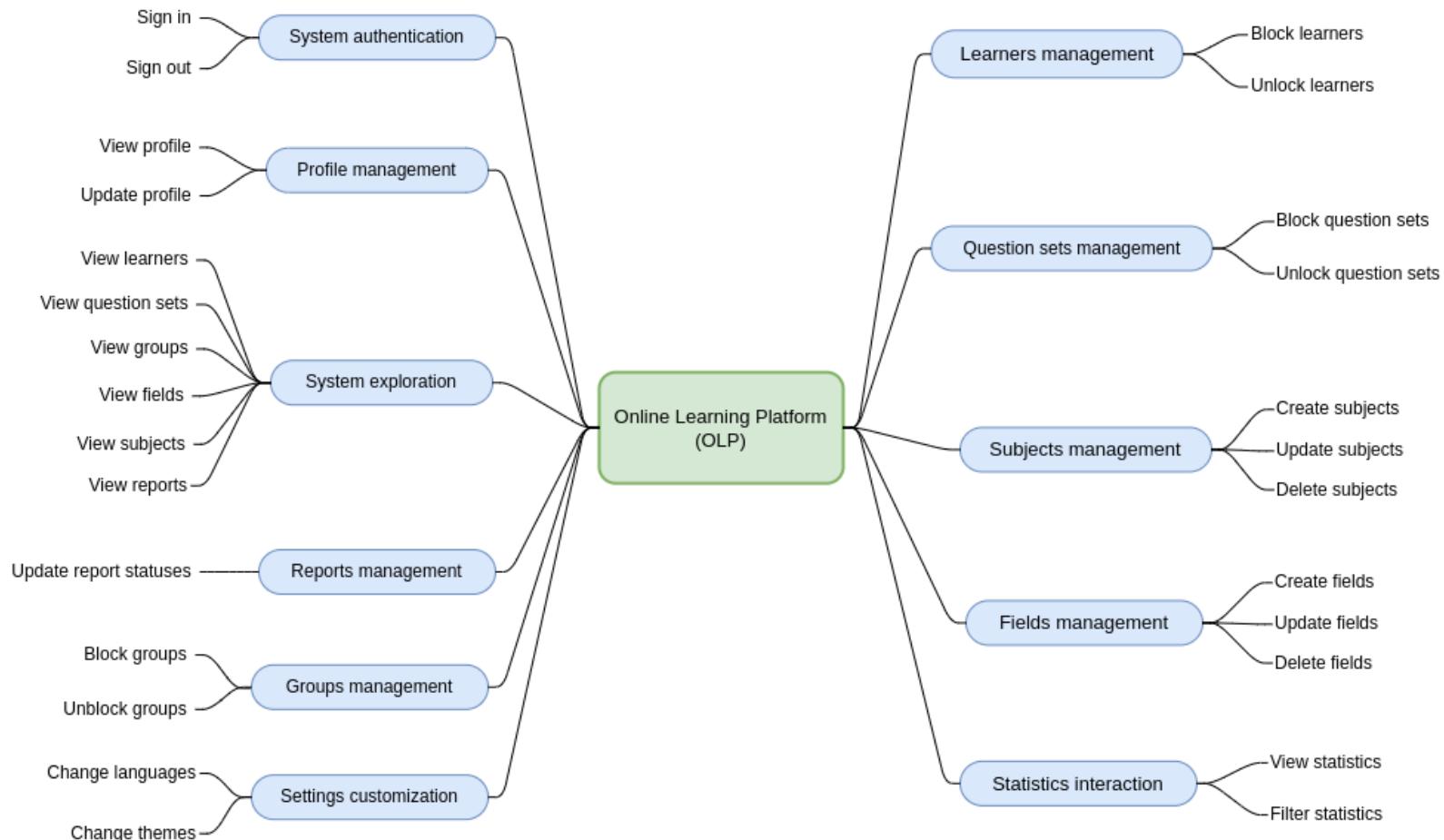


Figure 6.1.3: Administrator main features⁹

⁹ [admin-mind-map.drawio.png](#)

6.2 Limitations & Exclusions

LI - 1: The system will not include real-time, interactive tutoring features.

LI - 2: The system will not offer personalized, AI-based learning experiences that adjust to each user's needs.

LI - 3: The system is not intended to replace traditional education or provide recognized qualifications.

II. Project Management Plan

1. Overview

1.1 Scope & Estimation

#	WBS Item	Complexity	Est. Effort (man-days)
1	<i>Initiation</i>		11
1.1	Define project goals	Medium	3
1.2	Conduct research on existing systems	Medium	2
1.3	Build project charter	Medium	2
1.4	Kick-off meeting	Simple	0.5
1.5	Assign member roles and responsibilities	Simple	0.5
1.6	Define project risks	Complex	3
2	<i>Planning</i>		8
2.1	Define project scope	Medium	1
2.2	Develop project schedule	Medium	2
2.3	Resource and technology planning	Simple	1
2.4	Create risk management plan	Medium	2
2.5	Create training plan	Medium	1
2.6	Create communication plan	Simple	1
3	<i>Analyst and Design</i>		23
3.1	Define system actors	Medium	1
3.2	Conduct user requirement analysis	Medium	2
3.3	Determine features to developed	Medium	2
3.4	Design system architecture	Complex	2
3.5	Design database	Complex	5
3.6	Design website's prototypes	Complex	8

3.7	Review prototypes with team	Simple	3
4	<i>Documentation</i>		20
4.1	Develop software introduction	Medium	1
4.2	Develop project management plan	Medium	2
4.3	Develop Software requirement specification	Complex	6
4.4	Develop Software design description	Complex	6
4.5	Develop software testing documentation	Medium	2
4.6	Develop release package and user guide documentation	Medium	3
5	<i>Implement</i>		23
5.1	Create database	Medium	1
5.2	Implement back-end coding	Complex	10
5.3	Implement front-end coding	Complex	12
6	<i>Testing and maintenance</i>		10
6.1	Perform software testing	Medium	3
6.2	Maintain and update software and project document	Medium	3
7	<i>Closing</i>		3
7.1	Review project final report	Medium	1
7.2	Review project source code	Simple	0.5
7.3	Prepare a product demo video	Medium	1
7.4	Prepare for presentation	Simple	0.5

Total Estimated Effort (man-days) **98**

1.2 Project Risks

#	Risk Description	Impact	Possibility	Response Plans
1	Scope creeps	Delay project progress. Reduced deliverables quality. Impact allocated budget and time. Potential team conflicts.	Low	Create a detailed scope statement for project deliverables, timelines. Perform a measure to evaluate and approve changes requests.
2	Team member inefficiency	Delay project progress. Potential team conflict.	Medium	Ensure each member understands their roles and responsibilities. Host meeting regularly to address any performance issues.
3	The project falls behind schedule.	Impact allocated budget and time. Delay project progress. Dissatisfied stakeholders.	High	Monitor project progress regularly to spot any early potential deadlines. Develop a list of prioritised tasks based on their criticality to the project.
4	Changes in technology or system architecture	Create potential compatibility issues. Impact schedule.	Low	Stay informed about changes in the official document of chosen technologies. Conduct meetings to evaluate the impacts. Develop a contingency plan to migrate to new changes. Regularly testing and checking for quality performance.
5	Unforeseen technical problem	Delay project progress. Increased stress for team members.	Medium	Develop a contingency plan to address unforeseen technical problems. Allocate time for troubleshooting and debugging sessions.

2. Management Approach

Agile is a software development philosophy that emphasizes iterative development, customer collaboration, and continuous improvement. It prioritizes flexibility and adaptability, acknowledging that requirements and solutions often evolve throughout the project lifecycle. This approach contrasts with traditional waterfall methodologies, which follow a linear sequence of phases, often leading to inflexible plans and delayed delivery of value. Agile encourages frequent feedback loops, allowing teams to respond quickly to changes and deliver value incrementally.

Scrum is a well-defined Agile framework that provides a structured yet flexible approach for the software development process using specific roles, events or artifacts. It involves roles such as Product owner, Scrum master and development team, these roles are well defined with responsibilities. Scrum offers a set of iterative processes, called Sprints. For each sprint, the team will focus on delivering a set of functionalities, demonstrating it to the stakeholders and gathering feedback. Daily stand-up meetings foster communication and coordination, enabling the team to identify and address potential roadblocks.

Implementing the Agile/Scrum approach for an Online Learning Platform offers significant advantages. It enables the team to quickly adapt to changing user needs, such as incorporating new features based on user feedback or integrating emerging technologies in online education. The iterative approach allows for continuous testing and feedback from potential users, ensuring that the platform meets their learning needs and expectations. By embracing Agile/Scrum, the team can navigate uncertainties, respond effectively to evolving requirements, and ultimately deliver a successful and user-centric online learning platform.

2.1 Project Process

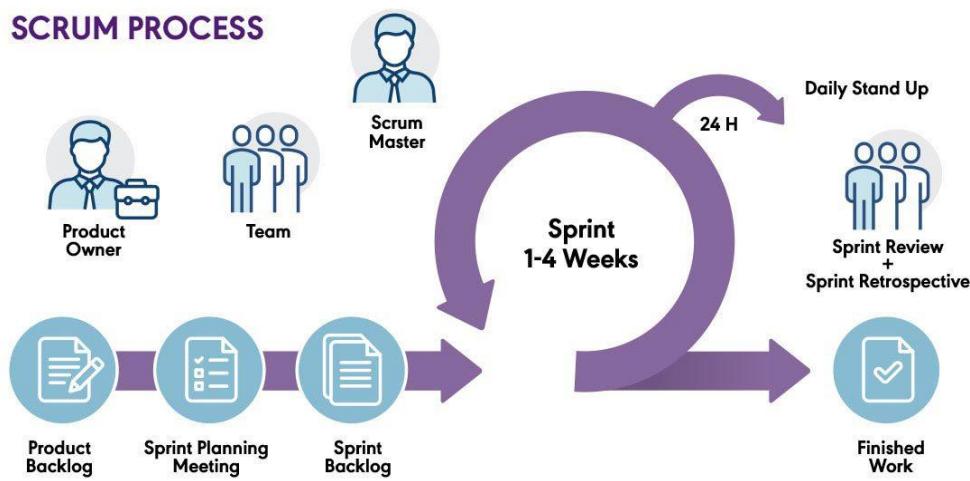


Figure 2.1: Scrum process¹⁰

By using Scrum, our software development process model consists of these following stages:

1. Project initiation:

- **Product vision:** Provide a shared understanding of the project overall scope and the ultimate goals among the whole team and the stakeholders. Ensure all the efforts are pointed towards a common goal.
- **Product backlog:** Create a list of features and functionalities based on stakeholder needs. Later priorities are based on system requirements and impacts.

2. Sprint planning (start of sprints):

- **Sprint backlog:** Create a list of priority items to be developed in the upcoming sprints by breaking the product backlog items into smaller chunks.
- **Sprint goal:** Define the goal of the sprint.
- **Estimate effort:** Estimate effort required for each sprint backlog.
- **Identify potential risks:** Discuss about any potential risks for the sprint.

¹⁰ <https://www.pm-partners.com.au/insights/the-agile-journey-a-scrum-overview/>

3. Sprint development:

- **Daily stand-up meetings:** A daily meeting is conducted, where team members will report task progress, discuss issues. The leader will continuously monitor the progress of the sprint, this helps identify potential issues early on.
- **Task integration:** Development team continue working on given tasks, while the leader assigns new tasks to keep up with the goals.
- **Continuous testing and quality assurance:** conduct a series test like unit testing, integration testing to ensure the reliability and quality of the system.

4. Sprint review:

- **Product demonstration:** Showcase the complete work of the sprint to the stakeholders.
- **Feedback gathering:** Gather feedback, improvement suggestions from the stakeholders for the next prints.
- **Adjust product backlog:** Based on the feedback, the team discusses adjusting the backlog for future sprints.

5. Sprint retrospective (end of sprints):

- **Team reflects:** The team will reflect on the team dynamics, communication and overall performance to discuss what went well and identify any challenges or areas for improvements.
- **Create action items:** The team generates specific, actionable items to address the identified areas of improvement.

By following these phases, the development process will ensure the stakeholders always receive high quality deliverables in a timely manner from the team, while the development team can effectively respond to any changing requirements from the stakeholders, further enhancing the collaboration between the team and the stakeholders.

2.2 Quality Management

Aligning with continuous project development, it is no denying that quality control is a crucial aspect. We prioritise quality assurance throughout the development cycle, so that values can be delivered consistently to the stakeholders. This process involves a high level of collaboration between the entire team and the stakeholders, ensuring high-quality standards. For every cycle, we regularly incorporate quality checks using such methods as reviewing and continuous testing.

Reviewing: reviewing here including a systematic review of source code and documentation to identify any potential issues. The process will be performed regularly throughout the development cycle inside each sprint, where each team member will demonstrate their working feature for the entire team to review and receive feedback. The working product is then demonstrated to the stakeholders at the end of each sprint to gather feedback from them. This helps the quality to be seen by any perspective.

- Advantages:
 - Team reviews provide an opportunity for multiple team members to examine the code, later identify potential issues or suggest improvements for the features.
 - By catching issues early on, reviewing helps to prevent defects from being introduced into the final products, reducing costly maintenance in upcoming sprints.
 - Product review from the stakeholders helps the team prioritise any quality attributes and verify that requirements are met.
 - Code review facilitates knowledge sharing within the team. Where one member can learn from other's things such as coding styles, best practices or problem-solving approaches. This concretes the bond between team members.
- Reason to implement:
 - This approach aligns with the collaborative nature of Agile, where progress is regularly monitored, and feedback is gathered for improvement.
 - Continuous review means higher chances for the product to meet the requirements.
 - Review is already a built-in phase of Agile, making this approach working seamlessly with any other phases.

Continuous testing: continuous testing is an ongoing process of executing tests either with or without automated tools throughout the whole software development cycle. This includes different types of testing, including unit tests, integration tests, user acceptance tests.

- Advantages:
 - Continuous testing helps to identify the issues early or during the development phases, rather than waiting until the later phases.
 - By continuously monitoring any upcoming issues, continuous testing helps the whole team prepare and reduces the risk of major problems later.
 - Reduced major problems means that team members can write high-quality and testable code.
 - Continuous testing encourages faster feedback to developers, allowing the issues to be fixed quickly.
- Reason to implement:
 - Aligns well with the iterative nature of Agile, where the final quality is not depicted by the current working product. Enable rapid feedback for quick adjustments.
 - Continuous testing means that the working product can meet the required standard and user expectations.
 - Continuous testing can be carried out automatically, enabling developers to focus on other tasks.

2.3 Training Plan

Training Area	Participants	Duration	Waiver Criteria
Figma	Design team, Front-end team	1 week	Mandatory
Next.JS	Front-end team	1 week	Mandatory
Express.js	Back-end team	1 week	Mandatory
MongoDB	All team members	1 week	Mandatory
Git, Github	All team members	1 week	Mandatory

3. Project Deliverables

#	Deliverable	Due Date	Deliverable Scope
1	Project Plan	11/01/2025	Plan project milestone
2	Report 1	11/01/2025	Evaluate requirements
3	Report 2	18/01/2025	Architecture Design
4	Report 3	08/02/2025	SRS Document design
5	Report 4	15/02/2025	Design class diagram and sequence diagram
6	Report 5	12/04/2025	Test document
7	Report 6	19/04/2025	User guide
8	Report 7	25/04/2025	Final document
9	Final Package	25/04/2025	Final Codes & documents, User manual
10	Code BE	25/04/2025	Source Code Backend
11	Code FE	25/04/2025	Source Code Frontend

4. Responsibility Assignments

4.1 Team & Structures

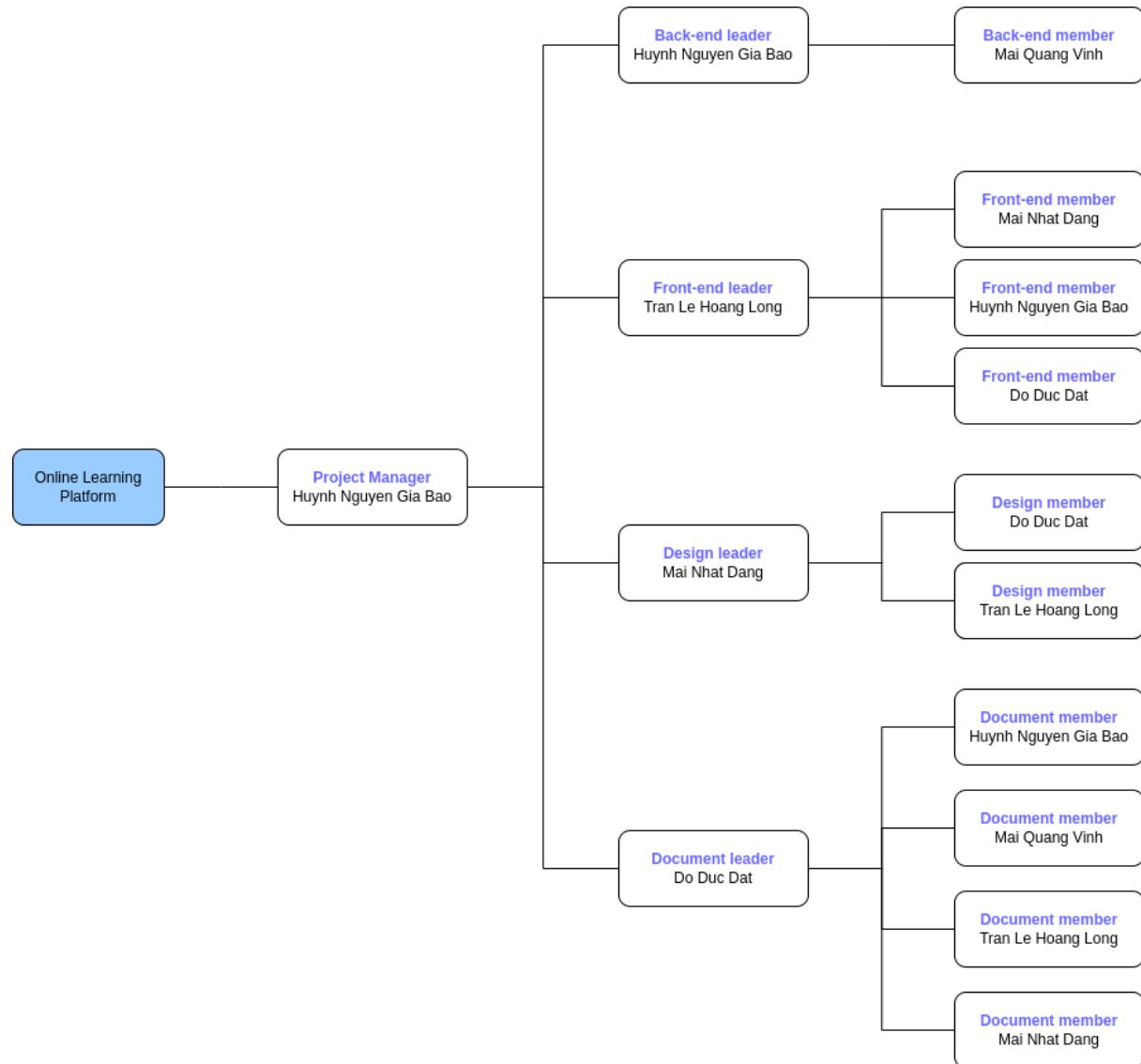


Figure 4.1: Team & Structure¹¹

¹¹ [Team-structure.drawio.png](#)

4.2 Role and Responsibility

Role	Responsibility
Project Manager	<ul style="list-style-type: none"> Leads overall project planning, execution, and monitoring, ensuring project goals and deadlines are met within budget. Communicates with stakeholders, manages risks and issues, and adjusts project plans as needed.
Design Leader	<ul style="list-style-type: none"> Leads the design process, including user research, wireframing, prototyping, and visual design. Ensures design consistency across all platforms and communicates design decisions effectively. Provides design feedback and guidance to developers.
Design Member	<ul style="list-style-type: none"> Create user-centric designs, including wireframes, mockups, and prototypes. Conduct user research and gather user feedback to inform design decisions. Collaborate with developers to ensure the accurate implementation of design specifications. Stay updated on the latest design trends and best practices.
Front-end leader	<ul style="list-style-type: none"> Guides front-end development, ensuring UI/UX consistency and adherence to best practices. Defines front-end architecture, participates in code reviews, and mentors junior front-end developers. Resolves technical challenges and ensures timely delivery of front-end features.
Front-end member	<ul style="list-style-type: none"> Develop, test, and maintain high-quality front-end code. Participate in code reviews and provide constructive feedback to peers. Collaborate with designers and back-end developers to ensure seamless integration. Stay updated on the latest front-end technologies and best practices.
Back-end leader	<ul style="list-style-type: none"> Guides back-end development, ensures the stability and performance of server-side systems. Defines back-end architecture, participates in code reviews, and mentors junior back-end developers. Resolves technical challenges and ensures timely delivery of back-end features.
Backend member	<ul style="list-style-type: none"> Develop, test, and maintain robust and scalable back-end systems. Participate in code reviews and provide constructive feedback to peers. Collaborate with front-end developers and ensure seamless integration between front-end and back-end systems. Stay updated on the latest back-end technologies and best practices.
Document leader	<ul style="list-style-type: none"> Oversees the creation and maintenance of all project documentation, including user manuals, technical documentation, and release notes. Ensures documentation is accurate, up-to-date, and easily accessible to all team members. Guides and mentors other team members in contributing to documentation efforts.
Document member	<ul style="list-style-type: none"> Contribute to the creation and maintenance of all project documentation. Ensure the accuracy and completeness of all documentation. Assist in the creation of user manuals, technical documentation, and release notes. Stay updated on best practices for technical documentation.

5. Project Communications

5.1 Communication Plan

Communication Item	Who/ Target	Purpose	When, Frequency	Type, Tool, Method(s)
FE meeting	BaoHNG, DangMN, LongTLH, DatDD	Notify about UI changes, resolve front-end-related issues.	Whenever front end has update	Discord, Google meet, figma
BE meeting	BaoHNG, DatDD, VinhMQ	Notify about API changes, resolve back-end-related issues.	Whenever back end has update	Discord, Google meet
Weekly meeting	All team, supervisor	Report project status, exchange information with the supervisor, identify issues.	Every Tuesday	Microsoft team
Intra-team meetings	All team	Resolve issues from both teams, report task status to the leader. Assign new tasks to members.	Daily or when notified by the leader	Discord, Google meet, Excel

5.2 External Interface

a. FU Contacts

Function	Contact Person (name, position)	Contact address (email, telephone)	Responsibility
Supervisor	Le Thi Phuong Dung	DungLTP13@fe.edu.vn 0901984504	- Provide document template - Instruct to project team - Review deliverables - Supervise project status

6. Configuration Management

6.1 Document Management



Figure 6.1.1: Google Docs¹²

Google Docs is a powerful online word processor that offers a suite of collaborative features, making it an invaluable tool for managing project documents. Beyond its core word processing capabilities, Google Docs excels in two key areas: real-time editing and robust version tracking.

¹² <https://www.stickpng.com/img/icons-logos-emojis/iconic-brands/google-docs-logo-horizontal>

Real-time editing is a key feature of Google Docs' collaborative tools. Several users can edit a document at the same time, seeing each other's changes instantly. This removes the necessity for unnecessary email exchanges, making the review and approval process more efficient. Team members can work together smoothly, giving instant feedback and solving problems together. This interactive setting promotes a feeling of mutual ownership and facilitates effective communication within the team.

The cloud-based nature of Google Docs ensures that every change inside a document can be viewed anytime. Every edit made to a document is automatically saved, creating a detailed record of changes over time. This allows users to easily revert to previous versions, recover from accidental deletions, or compare different versions to identify changes or resolve conflicts. The version history feature provides a safety net, ensuring that no work is permanently lost and enabling seamless recovery from unforeseen issues. This robust version control system is essential for maintaining document integrity and ensuring that the latest version reflects the most accurate and up-to-date information.

These two features, real-time editing and robust version tracking, combined with other functionalities like comment threads, access control, make Google Docs a standout choice for collaborative document management in today's dynamic work environments.



Figure 6.1.2: Google Sheets¹³

Google Sheets is a multi-functional cloud-based tool that is used for document editing and collaboration. It is a part of Google Workspace and combines other Google services into a single workplace for data management and analysis. The user-friendly interface makes it easy for all users, whether beginners or advanced, to operate at basic data entry or more sophisticated procedures like multi-layered calculations.

With Google Sheets, multiple users can work on a document at the same time and all changes appear in real-time. It has an extensive collection of multi-layered formulas that allows for varying levels of data processing and analysis. Important information can be conditional formatted for easy accessibility and data validation makes sure there is consistency and accuracy. In addition, charts and graphs aid in data visualization, making it effortless to comprehend and present information. Also, being hosted on a cloud makes it easy for users to access their data from any device connected to the internet as it's automatically saved in real-time.

Google Sheets can effectively be used for note-taking by creating structured lists, tables, or even utilizing the "Notes" feature of individual cells. This allows for grouping information, assigning tags or categories, and even inserting links to external resources within the spreadsheet.

¹³ <https://worldvectorlogo.com/logo/google-sheets-full-logo-1>

In conclusion, Google Sheets is a versatile tool for both personal and professional use. Its collaborative nature makes it ideal for tasks ranging from simple note-taking and data organization to complex data analysis and real-time monitoring.

6.2 Source Code Management



Figure 6.2.1: Git¹⁴

Git is a fast and light-weight distributed version control system (VCS) that allows multiple developers to collaborate, track changes and manage source code on a project efficiently.

One of its key features is version controlling, this allows developers to track changes in files and directories inside a repository overtime. The ability to create a snapshot of the project at different times make it easier to review, compare, revert or merge any changes without having to manually send or modify others code like the traditional method. Its core feature of branching and merging also enables flexible collaboration and code synchronization.

Not only lightweight and secure, Git is also designed to be compatible with various operating systems like Windows, macOS or Linux. Git can perform seamlessly with a wide range of environments and tools. This cross-platform compatibility ensures that developers can work effectively regardless of their development setup. Emerge as an open-source project, Git since then has been continuously improving, with extensive documentation, contributed by a large and active community.

The combination of robust version controlling and high compatibility, along with its blazing-fast yet secure nature has enabled developers to release high-quality products with minimal risks.



Figure 6.2.2: GitHub¹⁵

Github is a centralized platform where developers store and manage code repositories and collaborate with other developers. Leverage from the core feature of Git, it also supports a high level of version

¹⁴ https://vi.wikipedia.org/wiki/T%C3%A1i%ADp_tin:Git-logo.svg

¹⁵ <https://pngimg.com/image/73357>

tracking, along with other features such as seamless collaboration and extensive project management tools.

Github facilitates seamless collaboration among developers. With features like pull requests where they can submit their changes and ask for a review by other developers before being merged to the main codebase. This improves code quality and reduces the risk of unwanted errors.

From its main target to promote ease of management, Github offers a wide range of project management tools, including issue management, milestones, project board and even automated code review. This allows teams to organize their work, track progress and prioritize tasks with a single button click.

With a robust version control system, powerful tools for collaboration, along with a large, active community of developers, Github empowers developers to work more efficiently, produce higher-quality codebase, and easily contribute to an innovative open-source ecosystem.

6.3 Tools & Infrastructures



Figure 6.3.1: Visual Studio Code¹⁶

Visual Studio Code (VS Code) is a lightweight, yet powerful code editor developed by Microsoft. Its versatility, and extensibility while offering a user-friendly interface has made it stand out among other text editors.

One of VS Code's key strengths is its code editing capabilities, including code completion, syntax highlighting, and debugging support for a wide range of programming languages. Features like code refactoring, code navigation, and integrated terminal access also enhance developer productivity. The vast marketplace of extensions that VS Code offers also allow developers to customize their environment in every aspect.

VS Code provides integration with Git, this helps developers manage their code changes directly from the editor, which further enhances collaboration by making version control more accessible and intuitive. VS Code's marketplace also offers various Git-related tools and services, further enhancing its capabilities for version control management.

¹⁶ <https://www.stickpng.com/img/icons-logos-emojis/tech-companies/visual-studio-code-full-logo>

VS Code is an outstanding code editor thanks to its exceptional code editing capabilities, seamless Git integration, and unlimited customization. These features, combined with its user-friendly interface and active community, make it a highly productive and versatile tool for developers of all skill levels.



Figure 6.3.2: Clerk¹⁷

Clerk is an all-in-one authentication and user management platform that simplifies the challenges of implementing secure user access in today's website and mobile applications. It provides a range of pre-built components and APIs that take care of the complexities of authentication workflows, allowing developers to concentrate on the main logic of their applications. This platform makes the development process smoother by offering strong, ready-to-use solutions for managing user identities.

One of Clerk's standout features is its flexible authentication options. It accommodates various authentication methods, including traditional password logins, social logins (like Google and GitHub), and password-less options using email and SMS one-time passcodes. This diverse approach not only improves user accessibility but also upholds strict security measures, such as multi-factor authentication, to protect user credentials.

In addition, Clerk goes beyond basic authentication by providing extensive user management features. These include tools for handling user profiles, controlling sessions, and implementing role-based access control, which allows for detailed management of user permissions. By offering these capabilities, Clerk enables developers to efficiently manage user accounts and ensure a secure application environment.

In summary, Clerk greatly reduces the challenges linked to manual project setup by delivering a streamlined, pre-configured authentication and user management solution. This platform eliminates the need for developers to build custom authentication systems, thus saving development time and minimizing potential security risks. Its integrated features and strong security protocols enhance the development lifecycle, making it an essential tool for modern software projects.

¹⁷ <https://konfigthis.com/sdk/clerk/>



Figure 6.3.3: Cloudinary¹⁸

Cloudinary is one of the third-party companies that have revolutionized the practice of how businesses and developers manage media assets, particularly videos and images, through a complete set of tools.

Cloudinary offers a robust cloud infrastructure, designed to simplify the complexity of handling media assets. Uploading of photos and videos is done easily, where they are kept securely and retrieved easily through straightforward API calls, guaranteed to be fast and reliable no matter the user's location, improving the user experience overall. Cloudinary's powerful APIs also has the capability of dynamic image processing so that developers may resize, crop, apply filters, and many other things in real time from within their app. This flexibility eliminates the need for manual image correction and normalizes media for different devices and screen sizes. Along with basic storage and manipulation, Cloudinary offers features such as automatic image optimization, responsive image serving, and digital asset management capabilities. These allow for efficient workflows, bandwidth conservation, and improved site and app performance.

By offering a full media management platform, Cloudinary allows for ease of doing business in creating compelling content without the need to be dependent on the technical intricacies of managing media on a self-managed level.

¹⁸ https://commons.wikimedia.org/wiki/File:Cloudinary_logo.svg

III. Software Requirement Specification

1. Overall description

Online Learning Platform is a free software system that is designed to help learners and educators create and use flashcards, study games, and other interactive learning tools. This system is created with the aim to replace traditional, often tedious, study methods with engaging and adaptable digital resources. In this system, users can create their own question sets or explore millions of pre-existing sets created by others. The platform offers various study modes, including flashcards, matching games, quizzes and so on to adapt to different learning styles. Furthermore, the system encourages users to share and collaborate on study sets, promoting a form of collaborative learning. Instructors can also use the system to create custom study materials for other learners, while the learners can keep track of their progress and contribute answers to the sets. The diagram below illustrates the entities mentioned above.

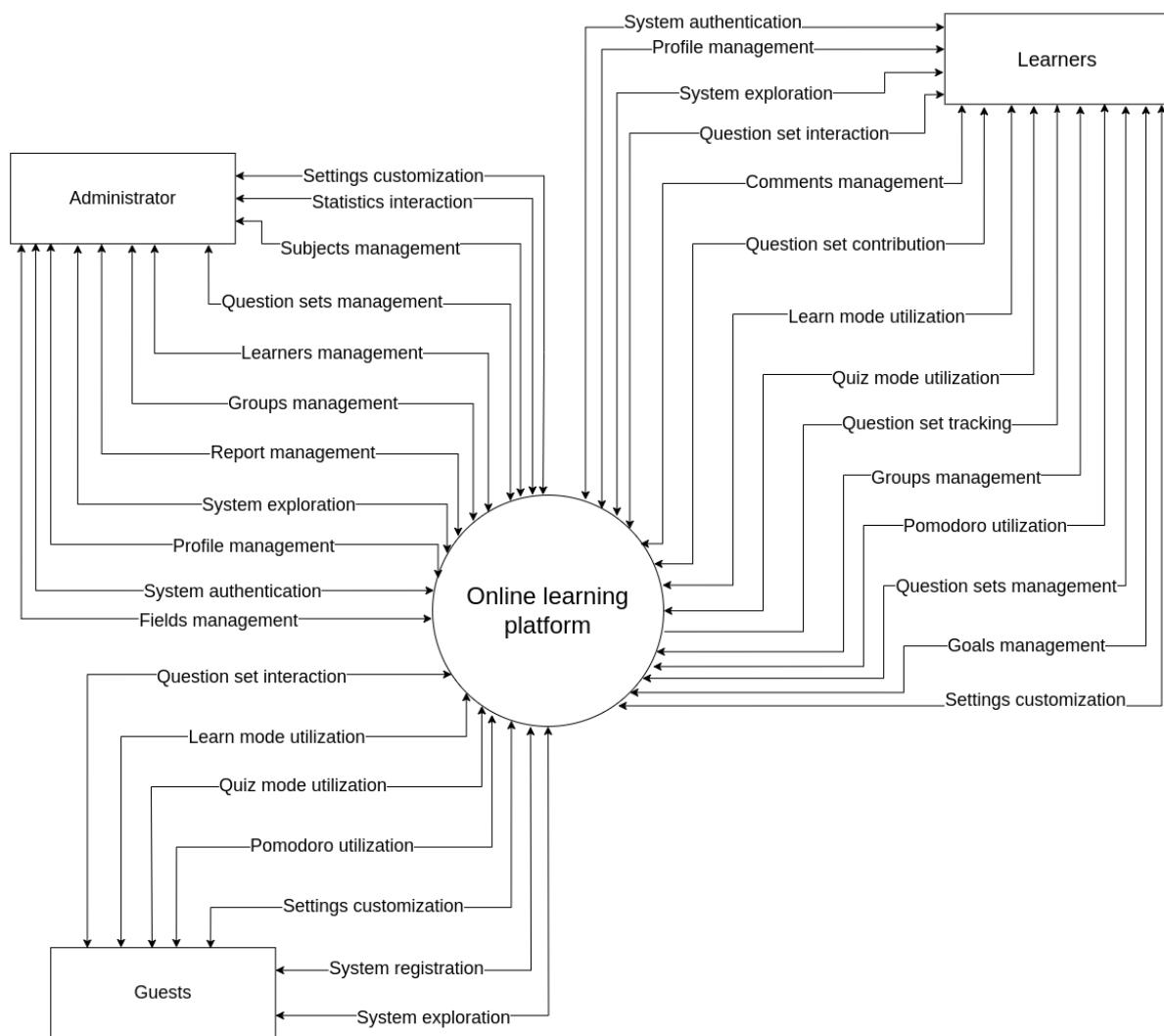


Figure 1: Project overview context diagram¹⁹

¹⁹ [Context-diagram.drawio.png](#)

2. User Requirements

2.1 Overview

a. Use Case Diagram

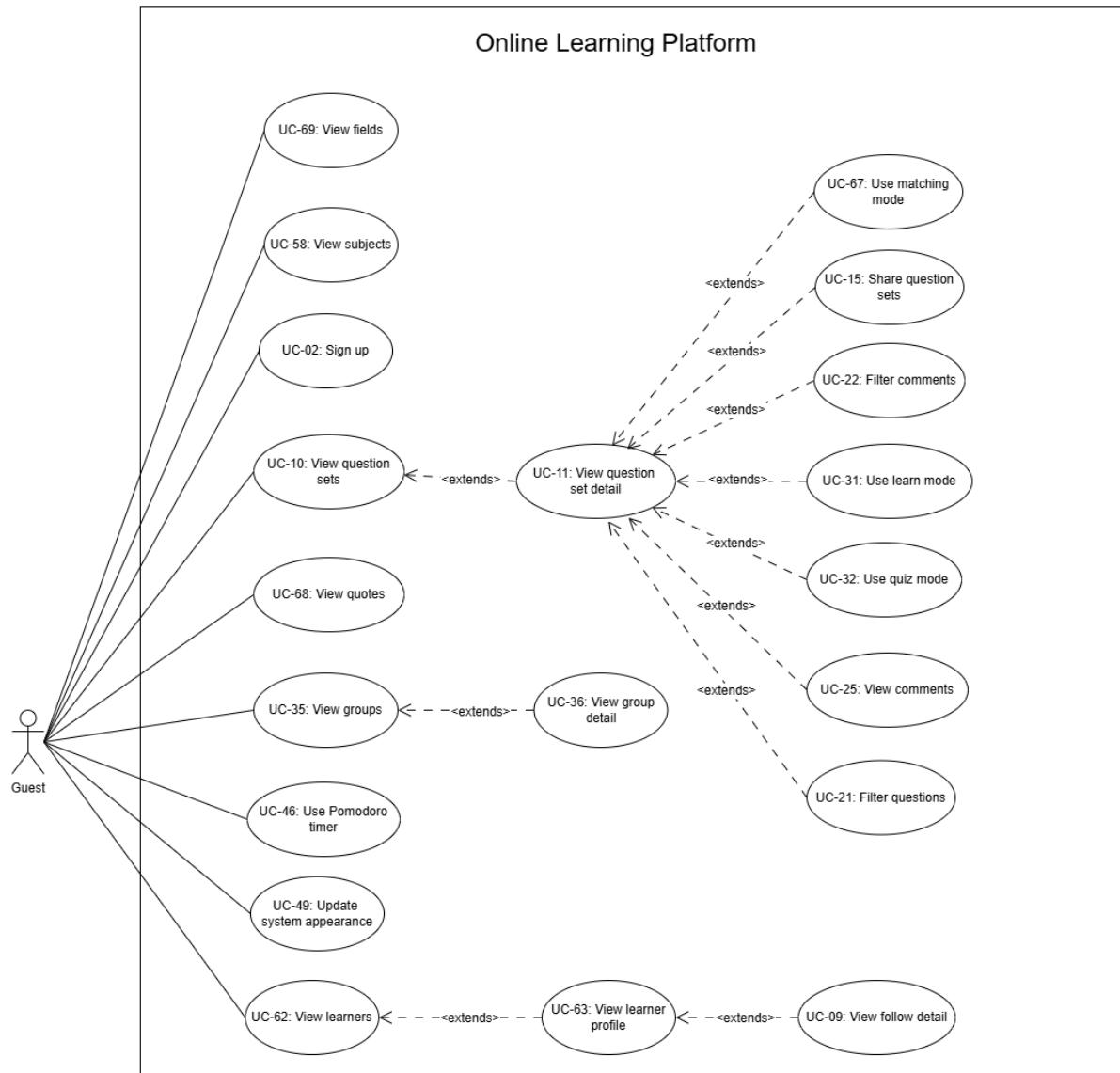


Figure 2.1.a.1: Guest's Website Use Case Diagram²⁰

²⁰ [guest-use-case.drawio.png](#)

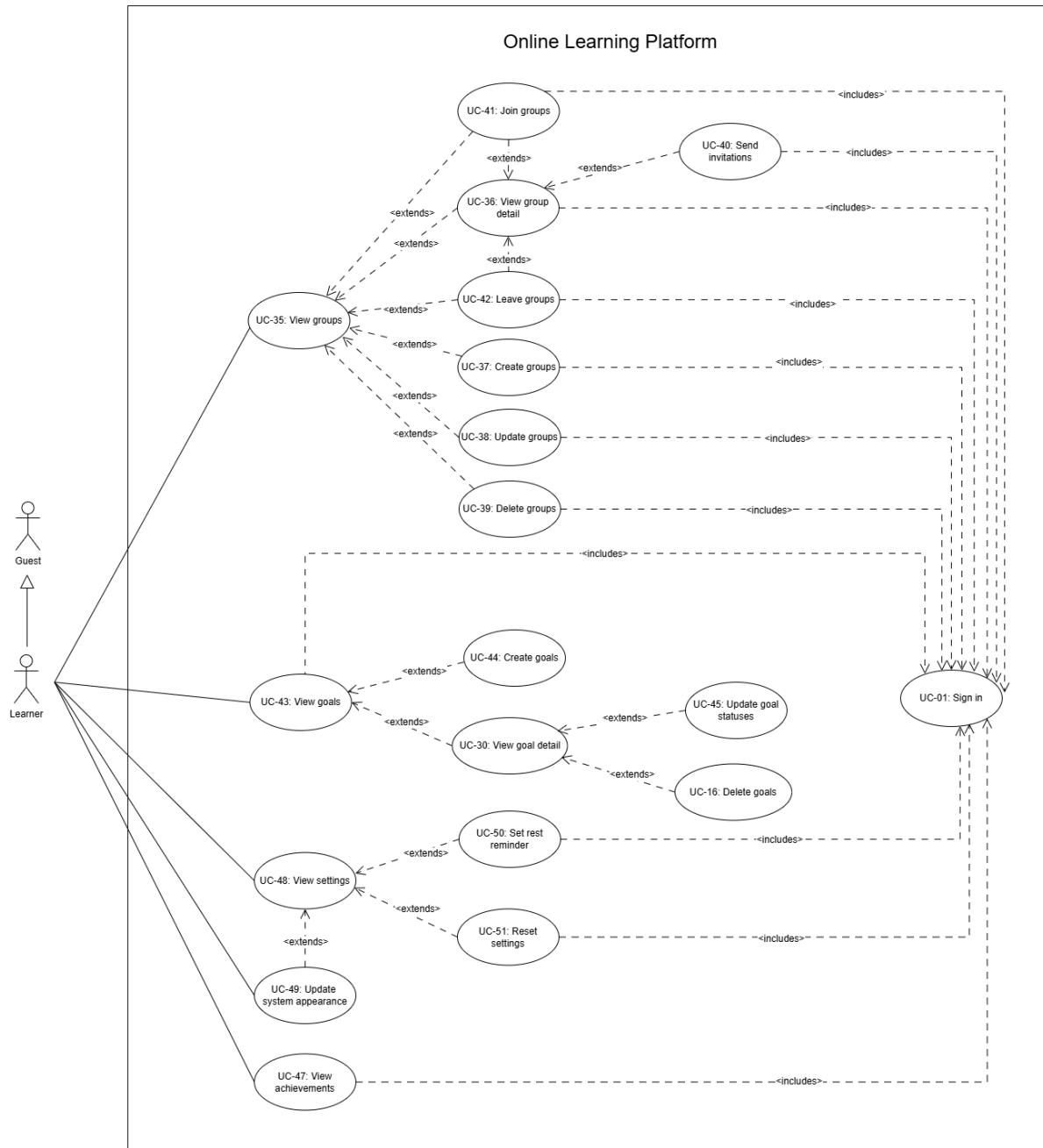


Figure 2.1.a.3: Learner's Website Use Case Diagram – 1²¹

²¹ [learner-use-case-1.drawio.png](#)

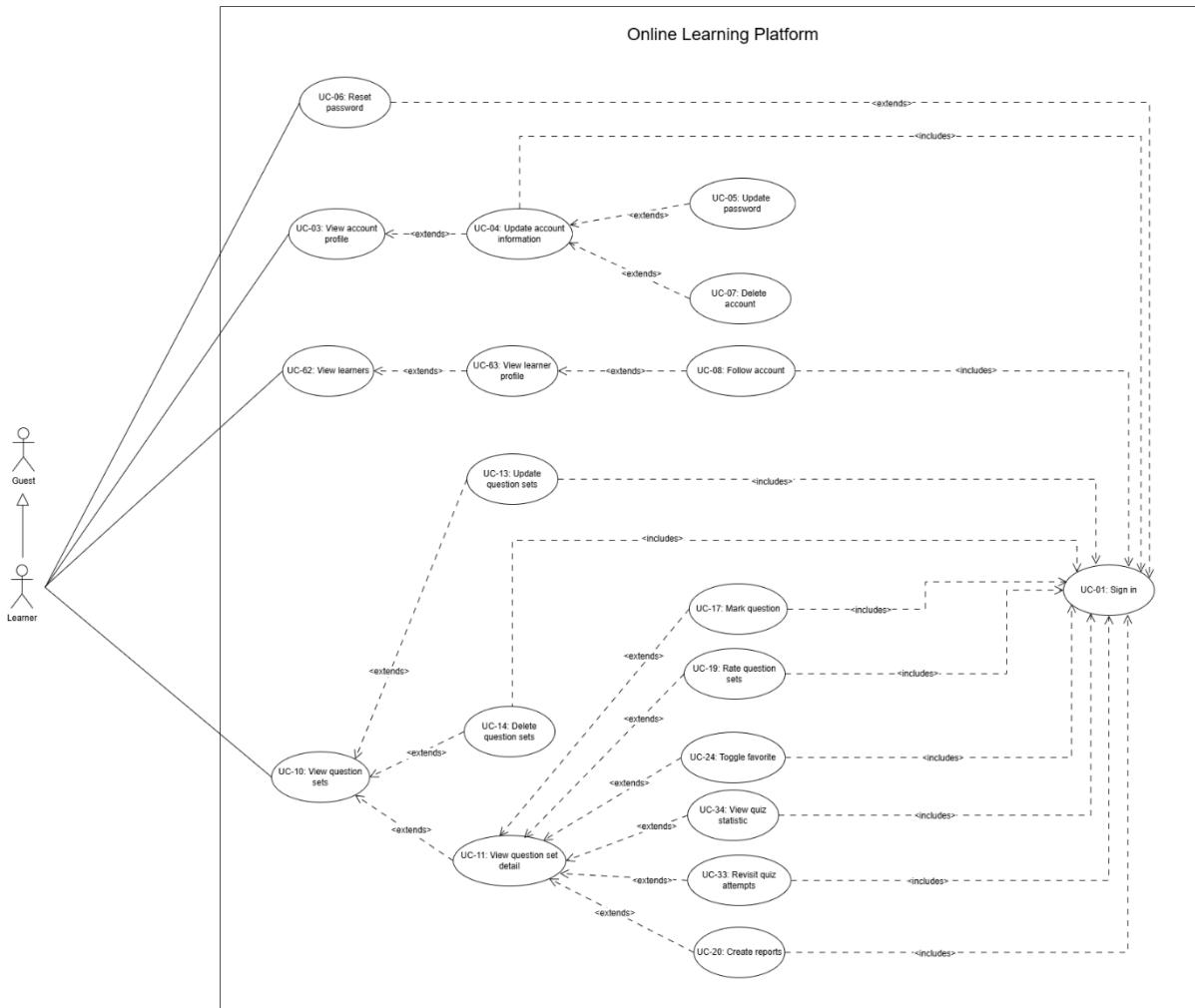


Figure 2.1.a.4: Learner's Website Use Case Diagram – 2²²

²² [learner-use-case-2.drawio.png](#)

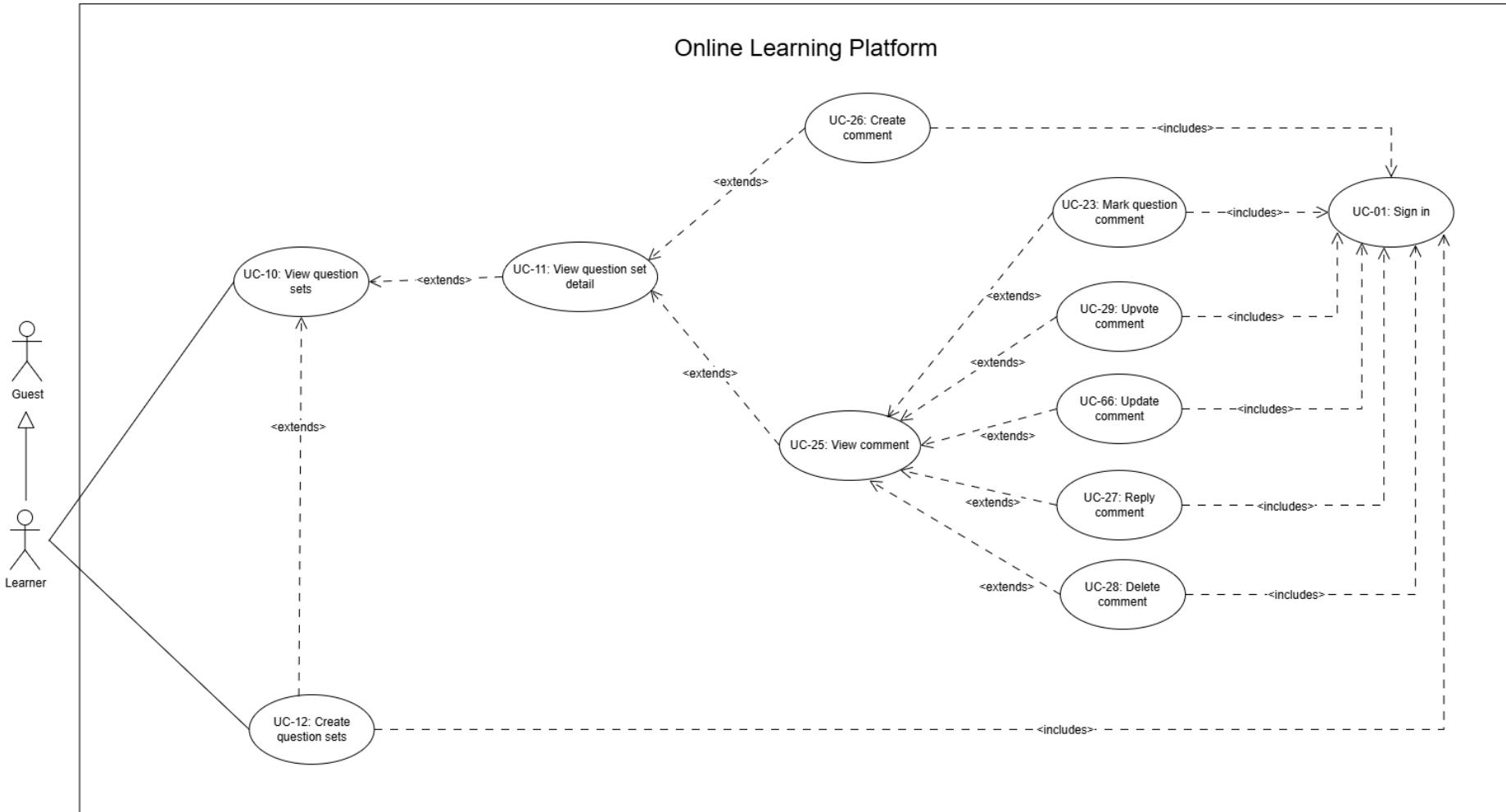


Figure 2.1.a.5: Learner's Website Use Case Diagram – 3²³

²³ [learner-use-case-3.drawio.png](#)

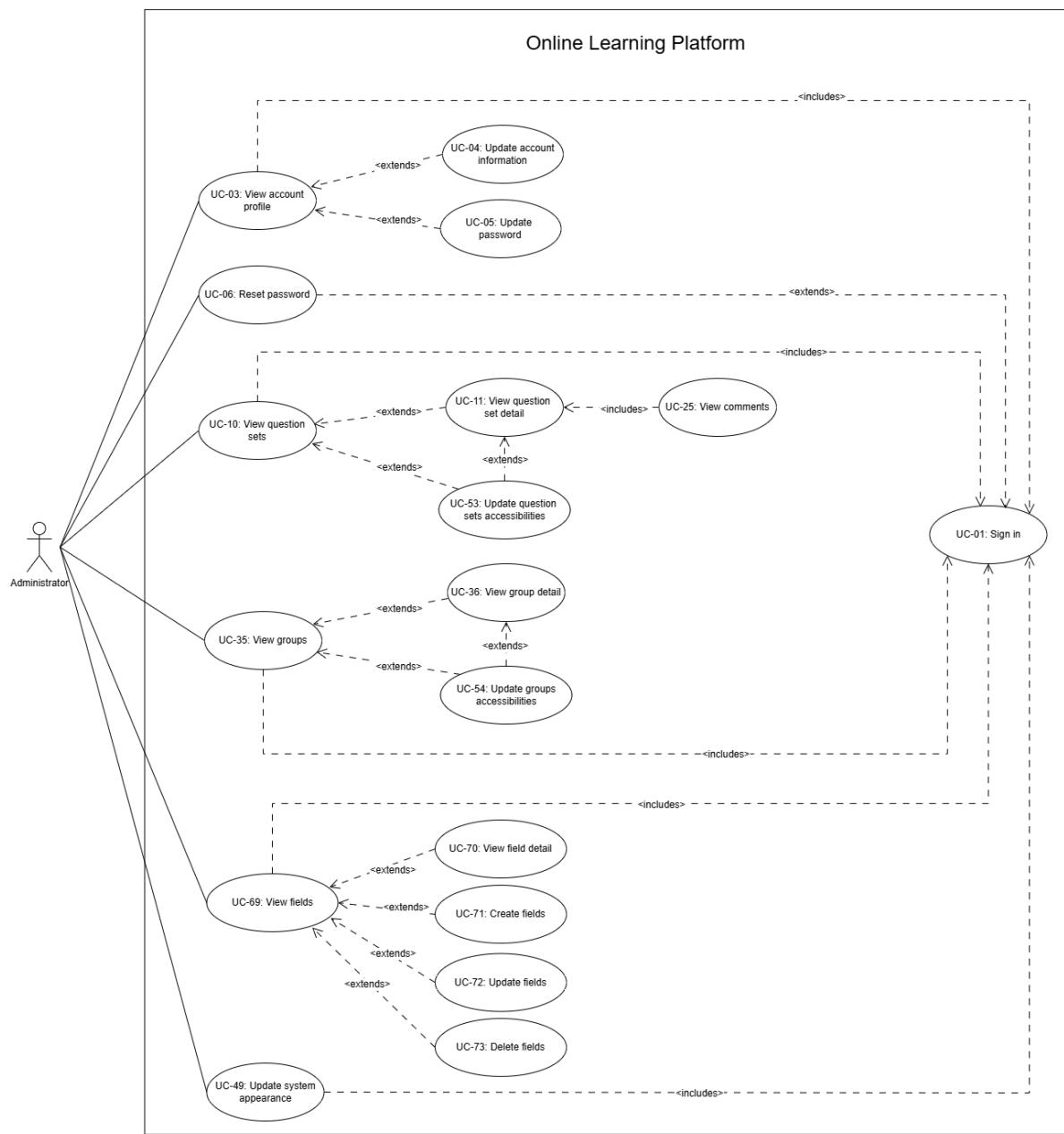


Figure 2.1.a.7: Administrator's Use Case Diagram – 1²⁴

²⁴ [admin-use-case-1.drawio.png](#)



Figure 2.1.a.8: Administrator's Use Case Diagram – 2²⁵

b. System Actors

#	Actor	Description
01	Guest	A guest is a user that has not signed up into the system. They have limited access to some features and information in the system.
02	Learner	A learner is a registered user who has full access to the system. They can use all the features that the system provided.
03	Administrator	An administrator is a registered user who has full access and control of the system and data. Administrator general roles include managing the platform usage and monitoring content.

²⁵ [admin-use-case-2.drawio.png](#)

c. Use Case List

ID	Use Case	Primary Actors	Secondary Actors
UC-01	Sign in	Learner, Administrator	
UC-02	Sign up	Guest	
UC-03	View account profile	Learner, Administrator	
UC-04	Update account information	Learner, Administrator	
UC-05	Update password	Learner, Administrator	
UC-06	Reset password	Learner, Administrator	
UC-07	Delete account	Learner	
UC-08	Follow account	Learner	
UC-09	View follow detail	Learner, Guest, Administrator	
UC-10	View question sets	Learner, Guest, Administrator	
UC-11	View question set detail	Learner, Guest, Administrator	
UC-12	Create question sets	Learner	
UC-13	Update question sets	Learner	
UC-14	Delete question sets	Learner	
UC-15	Share question sets	Learner, Guest	
UC-16	Remove goals	Learner	
UC-17	Mark questions	Learner	
UC-18	Export question sets	Learner	
UC-19	Rate question sets	Learner	

UC-20	Create report	Learner	
UC-21	Filter questions	Learner, Guest	
UC-22	Filter comments	Learner, Guest	
UC-23	Mark question comment	Learner	
UC-24	Toggle favorite	Learner	
UC-25	View comments	Learner, Guest, Administrator	
UC-26	Create comments	Learner	
UC-27	Reply comments	Learner	
UC-28	Delete comments	Learner	
UC-29	Upvote comments	Learner	
UC-30	View goal detail	Learner	
UC-31	Use learn mode	Learner, Guest	
UC-32	Use quiz mode	Learner, Guest	
UC-33	Revisit quiz attempts	Learner	
UC-34	View quiz statistics	Learner	
UC-35	View groups	Learner, Guest, Administrator	
UC-36	View group detail	Learner, Guest, Administrator	
UC-37	Create groups	Learner	
UC-38	Update groups	Learner	
UC-39	Delete groups	Learner	

UC-40	Send invitations	Learner	
UC-41	Join groups	Learner	
UC-42	Leave groups	Learner	
UC-43	View goals	Learner	
UC-44	Create goals	Learner	
UC-45	Update goal statuses	Learner	
UC-46	Use Pomodoro timer	Learner, Guest	
UC-47	View achievements	Learner	
UC-48	View settings	Learner	
UC-49	Update system appearance	Learner, Administrator	
UC-50	Set rest reminder	Learner	
UC-51	Reset setting	Learner	
UC-52	Update learner accessibilities	Administrator	
UC-53	Update question set accessibilities	Administrator	
UC-54	Update group accessibilities	Administrator	
UC-55	Update report statuses	Administrator	
UC-56	View statistics	Administrator	
UC-57	Filter statistics	Administrator	
UC-58	View subjects	Learner, guest, Administrator	
UC-59	Create subjects	Administrator	

UC-60	Update subjects	Administrator	
UC-61	Delete subjects	Administrator	
UC-62	View learners	Learner, guest, Administrator	
UC-63	View learner profile	Learner, guest, Administrator	
UC-64	View reports	Administrator	
UC-65	View report detail	Administrator	
UC-66	Update comment	Learner	
UC-67	Use matching mode	Learner, guest	
UC-68	View quote	Learner, guest	
UC-69	View fields	Learner, guest, Administrator	
UC-70	View field detail	Administrator	
UC-71	Create fields	Administrator	
UC-72	Update fields	Administrator	
UC-73	Delete fields	Administrator	
UC-74	View subject detail	Administrator	

2.2 User Authentication

2.2.1 Sign in

UC ID and Name:	UC-01 Sign in		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner, Administrator	Secondary Actors:	
Trigger:	1. User clicks on “Sign in” button		
Description:	Allow the user to provide their credential data, then get verified by the system to access the system.		
Preconditions:	1. The user must already have a valid account inside the system. 2. The user has not signed into the system		
Post-conditions:	1. The user is redirected to the home page		
Normal Flow:	A. User sign in with username 1. User clicks on the “Sign in” button. 2. User fills in the credential data. 3. User clicks on the “Continue” button. 4. The system sends the user to the homepage.		
Alternative Flows:	B. User sign in with email 1. User clicks on the “Sign in” button. 2. User fills in the credential data. 3. User clicks on the “Continue” button. 4. The system sends an OTP through user email. 5. User fills in the OTP. 6. User clicks on the “Continue” button. 7. The system sends the user to the homepage. C. User Sign in with Google account 1. User clicks on the “Sign in” button. 2. User clicks on the “Google” button. 3. The system redirects the user to the Google authentication page. 4. User chooses the account they want to sign in with. 5. The system sends the users to the homepage. D. User Sign in with invalid credentials 1. User clicks on the “Sign in” button. 2. User enter invalid username or password. 3. The system sends the request to Clerk server.		

	4. The system displays invalid data message
Exceptions:	1. User loses network connection. The system displays error message and cancels the request.
Priority:	High
Frequency of Use:	7 days after the user signed in or whenever after the user has signed out.
Business Rules:	BR-06, BR-08, BR-10, BR-11
Other Information:	N/A
Assumptions:	N/A

2.2.2 Sign up

UC ID and Name:	UC-02 Sign up		
Created By:	Huynh Nguyen Gia Bao	Date Created:	03/02/2025
Primary Actor:	Guest	Secondary Actors:	
Trigger:	1. User clicks on "Sign up" button		
Description:	Allow the user to create a new account inside the system by providing credential data.		
Preconditions:	1. User does not have an account inside the system. 2. User has not signed into the system.		
Post-conditions:	1. The user is redirected to the home page.		
Normal Flow:	<p>A. User sign up with email and username</p> 1. User clicks on the "Sign up" button. 2. User fills in the credential data. 3. User clicks on the "Continue" button. 4. The system changes into the OTP modal. 5. The system sends an OTP through user email. 6. User fills in the OTP. 7. User clicks on the "Continue" button. 8. The system sends the user to the homepage.		
Alternative Flows:	<p>B. User sign up with Google</p> 1. User clicks on the "Sign up" button. 2. User clicks on the "Google" button 3. The system redirects to the Google authentication page.		

	<p>4. User chooses the account they want to sign in with.</p> <p>5. User enters their username.</p> <p>6. User clicks on the “Continue” button.</p> <p>7. The system redirects the user to the homepage.</p> <p>C. User Sign up with invalid data</p> <p>1. User clicks on the “Sign up” button.</p> <p>2. User enters invalid data</p> <p>3. The system sends the data to clerk server.</p> <p>4. The system displays invalid data message to the user.</p>
Exceptions:	1. User loses network connection. The system displays error message and cancels the request.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-06, BR-10, BR-11
Other Information:	N/A
Assumptions:	N/A

2.3 Profile Management

2.3.1 View account profile

UC ID and Name:	UC-03 View account profile		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner, Administrator	Secondary Actors:	
Trigger:	1. User clicks on the “Profile” button.		
Description:	Allow the user to view their account information		
Preconditions:	1. The user must already have an account inside the system. 2. The user has signed in to the system.		
Post-conditions:	1. The user is redirected to the Account page. 2. The user views the account data displayed by the system.		
Normal Flow:	1. User clicks on the “Profile” button. 2. The system redirects the user to the Account page. 3. The system fetches user data and displays them to the user.		

Alternative Flows:	A. User is currently at the Settings page. 1. User clicks on the “Update” button from the Settings page 2. (Continue from step 3 of normal flow)
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	BR-08, BR-11
Other Information:	The administrator will not have alternative flows as the administrator does not have a Settings page.
Assumptions:	N/A

2.3.2 Update account information

UC ID and Name:	UC-04 Update account information		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner, Administrator	Secondary Actors:	
Trigger:	1. User clicks on the “Edit profile” button.		
Description:	Allow users to update their account information such as personal data, status, field of interest.		
Preconditions:	1. User is already at the Account page. 2. User has signed in to the system.		
Post-conditions:	1. The system updates the user account information with new data.		
Normal Flow:	A. Users change their personal data. 1. User clicks on the “Edit profile” button. 2. User field in their desired account information. 3. User clicks on the “Update” button. 4. The system verifies the data and sends it to the server. 5. The system returns a popup to notify the users. 6. The user views new data displayed on the Account page.		
Alternative Flows:	B. Users change their status. 1. User clicks on the plus icon next to the profile image. 2. User types in the status of desired into the status modal. 3. User clicks on the update button. 4. System changes the status and displays to the users.		

	<p>C. Users update their field of interest.</p> <ol style="list-style-type: none"> 1. User clicks on the brush icon inside the Account page. 2. User chooses subjects from the dropdown. 3. User clicks on the update button. 4. System updates the subjects and displays them to the users. <p>D. Users enter invalid data</p> <ol style="list-style-type: none"> 1. User clicks on the update button 2. User input invalid data 3. User clicks on the update button. 4. The system sends the request to the server 5. The system displays invalid data to the user <p>E. Users choose invalid file format for profile image.</p> <ol style="list-style-type: none"> 1. User clicks on the profile image. 2. User chooses incorrect file format. 3. User clicks on the select button. 4. The system sends the data to the clerk server. 5. The system displays invalid data message to the user.
Exceptions:	1. User loses network connection. The system displays error message and cancels the request.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-08, BR-11
Other Information:	The Administrator will only be able to update their account information data, the alternative flows will not be applied.
Assumptions:	N/A

2.3.3 Update password

UC ID and Name:	UC-05 Update password		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner, Administrator	Secondary Actors:	
Trigger:	1. User clicks on the “Edit profile” button from the Account page. 2. User goes to the “Security” tab.		

Description:	Allow users to change their credential password.
Preconditions:	1. User has signed in to the system. 2. User is currently inside the Account page
Post-conditions:	1. System update the password with the new one.
Normal Flow:	1. User clicks on the update password button from the Account page. 2. User inputs the old password. 3. User inputs the new password. 5. User clicks on the update button. 6. The system updates the new password for the user.
Alternative Flows:	<p>A. User enter invalid password</p> 1. User clicks on the update password button 2. User inputs incorrect old password. 3. User inputs incorrect new password. 4. User clicks on the update button. 5. The system sends the data to the clerk server. 6. The system displays the invalid data message. <p>B. User cancel updating password</p> 1. (Start from step 1 to step 3 of the normal flow) 2. User clicks on the cancel button 3. The system hide the modal and cancel the request.
Exceptions:	1. User loses network connection. The system displays error message and cancels the request.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-08, BR-10, BR-11
Other Information:	N/A
Assumptions:	N/A

2.3.4 Reset password

UC ID and Name:	UC-06 Reset password		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Learner, Administrator	Secondary Actors:	

Trigger:	1. User clicks on the “Forgot password” button on the Sign in modal
Description:	Allow users to reset their password if they forgot the current one.
Preconditions:	1. User has not signed in to the system. 2. Users must have a valid account inside the system.
Post-conditions:	1. The user is signed in and redirected to the Home page.
Normal Flow:	<p>1. User clicks on the “Forgot password” button on the Sign in modal.</p> <p>2. The system changes into Forgot password modal.</p> <p>3. User clicks on the “Reset password” button.</p> <p>4. The system changes into the OTP modal.</p> <p>5. The system sends an OTP code to the user email.</p> <p>6. User enters the OTP code.</p> <p>7. User change into the Reset password modal.</p> <p>8. The user enters the new password.</p> <p>9. The user clicks on the “Reset password”.</p> <p>10. The system verifies the password and redirects the user to the home page.</p>
Alternative Flows:	<p>A. User enter invalid OTP code</p> <p>1. (Start from step 1 to step 4 of the normal flow)</p> <p>2. User enters incorrect OTP code.</p> <p>3. The systems send the data to the clerk server</p> <p>4. The system displays incorrect data message to the user.</p> <p>B. User enter invalid password</p> <p>1. (Start from step 1 to step 8 of the normal flow)</p> <p>2. User enters invalid password.</p> <p>3. The system displays invalid data message to the user.</p>
Exceptions:	1. Users lose network connection. The system displays error message and cancels the request.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-10, BR-11
Other Information:	N/A
Assumptions:	N/A

2.3.5 Delete account

UC ID and Name:	UC-07 Delete account		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the “Delete account” button.		
Description:	Allow users to permanently remove their account from the system.		
Preconditions:	1. User has signed in to the system. 2. Users must have a valid account inside the system.		
Post-conditions:	1. user is redirected to the Landing page.		
Normal Flow:	1. User goes to the settings page 2. User clicks on the delete account button. 3. User clicks on the delete button inside the confirm popup. 4. The system sends the request to the server. 5. The system redirect the user to the landing page.		
Alternative Flows:	A. User cancel the deletion 1. User goes to the settings page. 2. User clicks on the delete account button. 3. User clicks on the cancel button inside the confirm popup 4. The system hide the popup and cancel the request.		
Exceptions:	1. User loses network connection. The system displays error message		
Priority:	High		
Frequency of Use:	Low		
Business Rules:	BR-24		
Other Information:	N/A		
Assumptions:	N/A		

2.4 Question Set Management

2.4.1 View question sets

UC ID and Name:	UC-10 View question sets		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner, guest, Administrator	Secondary Actors:	
Trigger:	1. User goes to the Home page		
Description:	Allow users to view available question sets created by other learners.		
Preconditions:	N/A		
Post-conditions:	1. The system displays the question sets to the user.		
Normal Flow:	1. User goes to the Home page. 2. The system fetches the data from the server. 3. The system displays data to the user.		
Alternative Flows:	<p>A. User search with keywords</p> 1. User puts the keyword into the search bar. 2. The system redirects the user to the Search result page. 3. The system fetches the data from the server based on the keyword provided. 4. The system displays data to the user. <p>B. User filter with Subjects</p> 1. User goes to the Explore page. 2. User clicks on the desired subjects. 3. The system redirects the user to the Search result page. 4. The system fetches the data from the server based on the subject chosen. 5. The system displays data to the user. <p>C. User view group detail</p> 1. User goes to the Group detail page. 2. The system fetches the data from the server based on the group. 3. The system displays data to the user. <p>D. Learner view favorited/ created question sets</p> 1. User goes to the sets page. 2. User clicks on the “Created” or “Favorited” tab button.		

	2. The system fetches the data from the server based on user choice. 3. The system displays data to the user.
Exceptions:	1. User loses network connection. The system displays error message
Priority:	High
Frequency of Use:	High
Business Rules:	BR-08, BR-13, BR-14, BR-17
Other Information:	N/A
Assumptions:	The system has question sets available for users to view.

2.4.2 View question set detail

UC ID and Name:	UC-11 View question set detail		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Learner, guest, Administrator	Secondary Actors:	
Trigger:	1. User clicks on the question set card		
Description:	Allow users to view the detail of a question set, including its description, questions, learners' comments...		
Preconditions:	N/A		
Post-conditions:	1. The system displays the question set related data to the user.		
Normal Flow:	1. User clicks on the question set card. 2. The system redirects the user to the question set detail page. 3. The system fetches the related data from the server. 4. The system displays the data to the user.		
Alternative Flows:	<p>A. User view question set with URL</p> 1. User enters the URL of the question set to the browser search bar. 2. (Continue from step 2 of the Normal flow) <p>B. User clicks on downloaded question set card</p> 1. User clicks on the question set card. 2. The system gets the data from the local storage. 3. The system displays the data to the user.		

	C. User view not-found question set 1. User enters the URL of the question set to the browser search bar. 2. The system redirects the user to the not found page.
Exceptions:	1. User loses network connection. The system displays error message
Priority:	High
Frequency of Use:	High
Business Rules:	BR-08, BR-13, BR-15, BR-16
Other Information:	learners can view the progress of themselves learning the set.
Assumptions:	N/A

2.4.3 Create question sets

UC ID and Name:	UC-12 Create question sets		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User goes to the create set page.		
Description:	Allow users to create their own question set, and decide whether to public the set to other users.		
Preconditions:	1. User has signed in to the system.		
Post-conditions:	1. The system redirects the user to the question set detail page of the newly created set.		
Normal Flow:	1. User goes to the create set page. 2. User fills in the data related to the set. 3. User clicks on the create button. 4. The system sends the data to the server. 5. The system redirects the user to the question set detail page.		
Alternative Flows:	A. User create question set by importing file 1. User goes to the create set page. 2. User clicks the import button. 3. User uploads the file with related data to the form. 4. (Continue from step 4 of the Normal flow) B. User left required field empty		

	<p>1. User goes to the create set page.</p> <p>2. User fills in the data but left the title field.</p> <p>3. User clicks on the create button</p> <p>4. The system displays missing data message to the user.</p> <p>C. User choose invalid file format</p> <p>1. User goes to the create set page.</p> <p>2. User clicks the import button</p> <p>3. User chooses an invalid file format</p> <p>4. User choose clicks the select button</p> <p>4. The system displays invalid data message to the user.</p>
Exceptions:	1. User loses network connection. The system displays error message
Priority:	High
Frequency of Use:	High
Business Rules:	BR-09, BR-13, BR-15
Other Information:	N/A
Assumptions:	The user knows the file format required by the system. User's uploaded file content matches the format required by the system.

2.4.4. Update question sets

UC ID and Name:	UC-13 Update question sets		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the update button.		
Description:	Allow users to update their created question set.		
Preconditions:	1. User has signed in to the system. 2. User's question sets are still active.		
Post-conditions:	1. The system redirects the user to the question set detail page of the modified question set.		
Normal Flow:	1. User clicks on the update button from the question set detail page. 2. The system redirects the user to the update question set page. 3. The system fetches the related data from the server. 4. The system displays the data to the user.		

	<p>5. The user changes their desired data.</p> <p>6. User clicks the update button.</p> <p>7. The system sends the modified data to the server.</p> <p>8. The system redirects the user to the question set detail page.</p>
Alternative Flows:	<p>A. User update a question from question set detail page</p> <p>1. User clicks on the edit icon inside the question card</p> <p>2. The system changes the card to form</p> <p>3. The user changes their desired data.</p> <p>4. User clicks the update button.</p> <p>5. The system sends the modified data to the server.</p> <p>6. The system updates the data inside the updated question card.</p> <p>B. User update a field to empty data</p> <p>1. (Start from step 1 to step 5 of the normal flow)</p> <p>2. User left a require field as empty</p> <p>3. User clicks on the update button</p> <p>4. The system displays missing data message to the user.</p>
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-13
Other Information:	N/A
Assumptions:	N/A

2.4.5 Delete question sets

UC ID and Name:	UC-14 Delete question sets		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner, Guest	Secondary Actors:	
Trigger:	1. User clicks on the delete icon.		
Description:	Allow users to delete their created question sets.		
Preconditions:	1. The user is currently using the website 2. The user has signed in to the website.		

	<p>3. The user is the author of the set to be deleted.</p> <p>4. The user is currently in the Sets page.</p>
Post-conditions:	1. The system displayed the updated list of question sets to the user.
Normal Flow:	<p>1. The user clicks on the delete icon on the question set card.</p> <p>2. The system shows the modal to confirm user action.</p> <p>3. The user clicks on the “Delete” button.</p> <p>4. The system sends the request to the server.</p> <p>5. The system displayed an updated question set list to the users.</p>
Alternative Flows:	<p>A. User cancels the deletion</p> <p>1. (Start from step 1 to step 2 of the normal flow)</p> <p>2. The user clicks on the cancel button.</p> <p>3. The system hide the modal and cancel the request.</p>
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-16
Other Information:	N/A
Assumptions:	N/A

2.4.6 Share question sets

UC ID and Name:	UC-15 Share question sets		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner, guest	Secondary Actors:	
Trigger:	1. User clicks on the “Share” button		
Description:	Allow users to share the question set as browser URL		
Preconditions:	1. The user is currently inside the question set detail page.		
Post-conditions:	1. The question set URL is copied into the user's system clipboard.		
Normal Flow:	<p>1. User clicks on the “Share” button.</p> <p>2. The system shows the modal displaying the question set URL.</p> <p>3. User clicks the copy icon.</p>		

	4. The question set URL is copied into the user's system clipboard.
Alternative Flows:	A. User share question set to group 1. User clicks on the "Share" button. 2. User clicks on the "Share to group" button. 3. User chooses the group to share. 4. The system sends a request to the server. 5. The system displays the updated list of question sets when the user goes to the group detail page.
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	Medium
Frequency of Use:	Medium
Business Rules:	BR-19, BR-14, BR-15, BR-16
Other Information:	N/A
Assumptions:	N/A

2.4.7 Export question sets

UC ID and Name:	UC-18 Export question set		
Created By:	Huynh Nguyen Gia Bao	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the "Export" button		
Description:	Allow users to export a question set to a file into their local machine		
Preconditions:	1. User has signed in to the system.		
Post-conditions:	1. The system downloads a file into the user system.		
Normal Flow:	1. User clicks on the "Export" button inside the question set detail page. 2. The system requests the server to send a file to the client. 3. The system receives file content and header from the server. 4. The system initiates the download. 5. The browser prompts the user to choose the download location or save automatically. 6. The browser downloads the file and saves it to the user's local machine.		

Alternative Flows:	N/A
Exceptions:	1. The user cancels the download. The browser interrupts the download process, and the user may have a partially downloaded file. 2. User loses network connection. The system displays error message.
Priority:	Medium
Frequency of Use:	Medium
Business Rules:	BR-15
Other Information:	N/A
Assumptions:	N/A

2.4.8 Rate question sets

UC ID and Name:	UC-19 Rate question sets		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on star icons		
Description:	Allow users to rate a question set by giving star		
Preconditions:	1. User has signed in to the system. 2. User has not rated a question set yet. 3. User is currently inside the question set detail page.		
Post-conditions:	1. The system displays the number of stars the user has chosen.		
Normal Flow:	1. User clicks on the star icons. 2. The system shows the confirmation modal. 3. User clicks the “Confirm” button. 4. The system sends the request to the server. 5. The system displays the number of stars rated by the user inside the question set detail page.		
Alternative Flows:	N/A		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	Medium		

Frequency of Use:	High
Business Rules:	N/A
Other Information:	N/A
Assumptions:	N/A

2.4.9 Toggle favourite

UC ID and Name:	UC-24 Toggle favorite		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner, Guest	Secondary Actors:	
Trigger:	1. User clicks on the heart icon.		
Description:	Allow users to add a question set to their favorite question sets.		
Preconditions:	1. User is currently using the website 2. User has signed in to the website.		
Post-conditions:	1. The system updates the heart icon's look to the user.		
Normal Flow:	1. User clicks on the heart icon inside the question set detail page. 2. The system sends the request to the server. 3. The system updates the heart icon's look to the user.		
Alternative Flows:	A. User remove the favourited question set from the list 1. User goes to the Sets page. 2. User clicks on the “Favorites” tab. 3. The system fetches the favorited question sets from the server. 4. The system displays the data to the user. 5. User clicks on the heart icon on the question set card. 6. The system sends the request to the server. 7. The system displays the updated list of favorited question sets to the user.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-18		

Other Information:	N/A
Assumptions:	N/A

2.4.10 Update question set accessibilities

UC ID and Name:	UC-53 Update question set accessibilities		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User clicks on the “Block”/“Unblock” button inside question set table row		
Description:	Allow the Administrator to block or unblock a question set created by learners.		
Preconditions:	N/A		
Post-conditions:	1. The system displays the status of the question set inside the table row.		
Normal Flow:	1. User clicks on the “Block” button on the question set table row. 2. The system sends the request to the server. 3. The system updates the “Block” button to “Unblock” button to the user.		
Alternative Flows:	A. The Administrator unlock the question set 1. User clicks on the “Unblock” button on the question set table row. 2. The system sends the request to the server. 3. The system updates the “Unblock” button to “Block” button to the user. B. The Administrator blocks the question set while resolving the report. 1. User clicks on the “Resolve” dropdown button. 2. User clicks on “Block question set” from the dropdown. 3. The system displays a confirmed modal. 4. User clicks on the “Resolve” button. 5. The system sends the request to the server. 6. The system displays a message to notify the user the report has been resolved.		
Exceptions:	1. Administrator clicks the “Cancel” button. The system cancels the action and hides the confirmation modal.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-08, BR-14, BR-15, BR-23		

Other Information:	N/A
Assumptions:	The Administrator acknowledge the reason why a question set should be block/unblock

2.5 Group Management

2.5.1 View groups

UC ID and Name:	UC-35 View groups		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Learner, guest, Administrator	Secondary Actors:	
Trigger:	1. User requests to view groups information (Administrator's groups page, search result page, learners' groups page)		
Description:	Allow users to view groups created by learners inside the system		
Preconditions:	N/A		
Post-conditions:	The system displays a list of groups to the user.		
Normal Flow:	1. User requests to view groups information 2. The system fetches data from the server. 3. The system displays a list of groups to the user.		
Alternative Flows:	A. User filters for learners in the search result page. 1. User input keyword into the search bar. 2. The system redirects the user to the search result page. 3. User clicks on the learner filter option on the sidebar. 4. (continue from step 2 of the normal flow) A. Learner is inside the groups page. 1. User goes to the groups page. 2. The system fetches data from the server. 3. The system displays a list of participating groups to the user.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-08, BR-13		

Other Information:	Learners' groups page will show groups they are participating in. While the Administrator's groups page will list all the groups created in the system.		
Assumptions:	N/A		

2.5.2 View group detail

UC ID and Name:	UC-36 View group detail		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Learner, guest, Administrator	Secondary Actors:	
Trigger:	1. User clicks on the group card		
Description:	Allow users to view the detail of a groups, including participated members, shared question sets		
Preconditions:	N/A		
Post-conditions:	1. The system displays the group related data to the user		
Normal Flow:	1. User clicks on the group card 2. The system redirects the user to the group detail page 3. The system fetches the related data from the server. 4. The system displays the data to the user		
Alternative Flows:	A. User view groups from URL. 1. User types the URL of the group to the browser search bar. 2. (Continue from step 2 of the Normal flow) B. User view non-existent groups from URL. 1. User types the URL of the group to the browser search bar. 2. The system redirects the user to the not found page.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-08, BR-13		
Other Information:	N/A		
Assumptions:	N/A		

2.5.3 Create groups

UC ID and Name:	UC-37 Create groups		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User goes to the create group page		
Description:	Allow users to create their own group to store question sets, including sharing the sets with other learners.		
Preconditions:	1. User has signed in to the system		
Post-conditions:	1. The system redirects the user to the group detail page of the newly created group		
Normal Flow:	1. User goes to the create group page. 2. User fills in the data related to the group. 3. User clicks on the “Create” button. 4. The system sends the data to the server. 5. The system redirects the user to the group detail page.		
Alternative Flows:	A. User create group with empty required field 1. User goes to the create group page. 2. User leave a require field. 3. The system disable the continue button until data is valid.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-14, BR-19		
Other Information:	N/A		
Assumptions:	N/A		

2.5.4 Update groups

UC ID and Name:	UC-38 Update groups		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	

Trigger:	1. User clicks on the “Edit” button.
Description:	Allow users to update their created groups, including assigning member permissions
Preconditions:	1. User has signed in to the system. 2. User's groups are still active. 3. The user is the leader of the group.
Post-conditions:	1. The system shows the message to notify the user about the changes.
Normal Flow:	1. User clicks on the “Edit” button from the group detail page. 2. The system redirects the user to the update group page. 3. The system fetches the related data from the server. 4. The system displays the data to the user. 5. User changes the desired data. 6. User clicks on the “Update button” 7. The system sends the modified data to the server. 8. The system shows the message to notify the user about the changes.
Alternative Flows:	A. Group leader removes the user from group 1. User clicks on the remove icon on the user card. 2. User clicks on the “Update” button. 3. The system sends a request to the server. 4. The system displays a message notifying the user that the member has been removed.
Exceptions:	1. User lose network connection. The system displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	BR-01, BR-20
Other Information:	N/A
Assumptions:	N/A

2.5.5 Delete groups

UC ID and Name:	UC-39 Delete groups		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	

Trigger:	1. User clicks on the “Delete” button
Description:	Allow group leaders to delete their created groups
Preconditions:	1. User has signed in to the system. 2. User’s groups are still active. 3. The user is the leader of the group.
Post-conditions:	1. The system redirects the user back to the groups page
Normal Flow:	1. User clicks on the “Edit” button from the group detail page. 2. The user clicks on the “Delete” button. 3. The system shows the modal to confirm user action. 4. The user clicks on the “Delete” button. 5. The system sends the request to the server. 6. The system redirects the user back to the groups page.
Alternative Flows:	N/A
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-01, BR-13, BR-20
Other Information:	N/A
Assumptions:	N/A

2.5.6 Send invitations

UC ID and Name:	UC-40 Send invitations		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User goes to the Invitation list tab.		
Description:	Allow group members to send invitations to non-group members.		
Preconditions:	1. User has signed in to the system. 2. User’s groups are still active. 3. User has permission to send invitations.		

Post-conditions:	1. The system displays the message notifying the user the invitation has been sent.
Normal Flow:	<ol style="list-style-type: none"> 1. User goes to the Invitation list tab. 2. User clicks on the “Invite” button. 3. User enters a user’s username. 4. User chooses a participated group name 5. User clicks on the “Invite” button. 6. The system sends the data to the server. 7. The system displays the message notifying the user the invitation has been sent.
Alternative Flows:	<p>A. User cancels the invitation</p> <ol style="list-style-type: none"> 1. (Start from step 1 to step 3 of the normal flow) 2. User clicks on the cancel button 3. The system hides the modal and cancels the request <p>B. User tries to invite a non-existent user</p> <ol style="list-style-type: none"> 1. (Start from step 1 to step 4 of the normal flow) 2. The system displays not found message to the user.
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	BR-03, BR-05, BR-19, BR-20
Other Information:	N/A
Assumptions:	N/A

2.5.7 Join groups

UC ID and Name:	UC-41 Join groups		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the “Join” button.		
Description:	Allow users to join a group, including joining with invitations.		
Preconditions:	1. User has signed in to the system. 2. User has not joined the group.		

Post-conditions:	1. The system redirects the user to the newly joined group.
Normal Flow:	<ol style="list-style-type: none"> 1. User clicks on a group card. 2. The system redirects the user to the group detail page. 3. User clicks on the “Join” button. 4. The system sends a request to the server. 5. The system changes the “Join” button to the “Leave” button.
Alternative Flows:	<p>A. User join with invitations</p> <ol style="list-style-type: none"> 1. User goes to the Invitation list. 2. The system fetches the data from the server. 3. The system displays the invitations to the user. 4. User clicks on the “Accept” button. 5. The system sends a request to the server. 6. The system updates the invitation list to the user. <p>B. User decline the invitation</p> <ol style="list-style-type: none"> 1. User goes to the Invitation list. 2. The system fetches the data from the server. 3. The system displays the invitations to the user. 4. User clicks on the “Decline” button. 5. The system sends a request to the server to delete the invitation. 6. The system updates the invitation list to the user.
Exceptions:	<ol style="list-style-type: none"> 1. The user leaves the invitation unread. The system automatically removes the invitation after 3 days. 2. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	BR-05, BR-19, BR-20
Other Information:	N/A
Assumptions:	N/A

2.5.8 Leave groups

UC ID and Name:	UC-42 Leave groups		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the “Leave” button.		
Description:	Allow users to leave the current group they are participating in.		
Preconditions:	1. The user has signed in to the system. 2. The user has joined the group.		
Post-conditions:	1. The system displays a message notifying the user that they have left the group.		
Normal Flow:	1. User goes to the group detail page. 2. User clicks on the “Leave” button. 3. User clicks on the “Leave” button from the modal. 4. The system sends a request to the server. 5. The system displays a message notifying the user that they have left the groups. 6. The system changes the “Leave” button to the “Join” button.		
Alternative Flows:	A. User clicks “Leave” from the group card 1. User clicks on the “Leave group” option. 2. User clicks on the “Leave” button from the modal. 3. The system sends a request to the server. 4. The system displays a message notifying the user that they have left the groups.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-01, BR-20		
Other Information:	N/A		
Assumptions:	N/A		

2.5.9 Update group accessibilities

UC ID and Name:	UC-54 Update group accessibilities		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User clicks on the “Block”/“Unblock” button inside group table row		
Description:	Allow the Administrator to block or unblock a group created by learners.		
Preconditions:	N/A		
Post-conditions:	1. The system displays the status of the group inside the table row.		
Normal Flow:	1. User clicks on the “Block” button on the group table row. 2. The system sends the request to the server. 3. The system updates the “Block” button to “Unblock” button to the user.		
Alternative Flows:	A. The Administrator unlock the group 1. User clicks on the “Unblock” button on the group table row. 2. The system sends the request to the server. 3. The system updates the “Unblock” button to the “Block” button to the user.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-08, BR-23		
Other Information:	N/A		
Assumptions:	N/A		

2.6 Goal Management

2.6.1 View goals

UC ID and Name:	UC-43 View goals		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	

Trigger:	1. User goes to the goals page
Description:	Allow users to view the goals created by themselves, this including searching and filtering for goals.
Preconditions:	1. User has signed in to the system.
Post-conditions:	1. The system displays a list of created goals to the user.
Normal Flow:	<ol style="list-style-type: none"> 1. User goes to the goals page. 2. The system fetches the data from the server. 3. The system displays a list of created goals to the user.
Alternative Flows:	A. A goal deadline is near or due <ol style="list-style-type: none"> 1. User goes to the Home page. 2. The system fetches the data from the server. 3. The system displays a row of due goals to the user.
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	BR-21
Other Information:	N/A
Assumptions:	N/A

2.6.2 View goal detail

UC ID and Name:	UC-30 View goal detail		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on goal card		
Description:	Allow users to view the details of a created goal, including current progress and deadline.		
Preconditions:	<ol style="list-style-type: none"> 1. User has signed in to the system. 2. User is currently inside the goals page. 3. Goals are available 		
Post-conditions:	1. The system displays goal data to user		

Normal Flow:	1. User clicks on goal card 2. The system fetches the data from the server. 3. The system displays goal data to the user.
Alternative Flows:	N/A
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-21
Other Information:	N/A
Assumptions:	N/A

2.6.3 Create goals

UC ID and Name:	UC-44 Create goals		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the “Create” button		
Description:	Allow user to create their own goals		
Preconditions:	1. User has signed in to the system.		
Post-conditions:	1. The system displays a list of goals to the user.		
Normal Flow:	1. User goes to the goals page. 2. User clicks on the “Create” button. 3. User fills in the necessary data. 4. User clicks on the “Create” button. 5. The system sends the data to the server. 6. The system displays the updated list of goals to the user.		
Alternative Flows:	A. User leave required field as empty. 1. (Start from step 1 to step 2 of the normal flow) 2. User leaves required field as empty 3. User clicks on the “Create” button 4. The systems displays missing data to the user.		

Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	BR-21
Other Information:	N/A
Assumptions:	N/A

2.6.4 Update goal statuses

UC ID and Name:	UC-45 Update goal statuses		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User chooses a status on a goal card.		
Description:	Allow users to update the status of a goal, including statuses like "Finished", "Not doing", which told the system to stop tracking progress of goals.		
Preconditions:	1. User has signed in to the system. 2. User has goals available. 3. User's goals are still being kept track.		
Post-conditions:	1. The system updates the list of goals to the user.		
Normal Flow:	1. User clicks on the edit icon on the goal card. 2. User clicks on a status. 3. The system sends a request to the server. 4. The system updates the list of goals to the user.		
Alternative Flows:	A. User updates goal progress 1. User clicks on the edit icon on the goal card. 2. User enter a value inside progress input 3. User clicks the "Update" button. 4. The system sends the data to the server. 5. The system displays the progress inside the goal card		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		

Frequency of Use:	Medium
Business Rules:	BR-21
Other Information:	Unless a goal is deleted, the updated goal detail is still visible to the author.
Assumptions:	N/A

2.6.5 Delete goals

UC ID and Name:	UC-16 Delete goals		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	User clicks on the delete button inside goal detail modal		
Description:	Allow users to delete their created goals.		
Preconditions:	1. User has signed in to the system. 2. User has goals available.		
Post-conditions:	1. The system updates the list of goals to the user.		
Normal Flow:	1. User clicks on the delete icon. 2. User clicks on the “Delete” button inside the confirm modal. 3. The system sends a request to the server. 4. The system updates the list of goals to the user.		
Alternative Flows:	N/A		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	Medium		
Frequency of Use:	High		
Business Rules:	BR-21		
Other Information:	Unless a goal is deleted, the canceled goal detail is still visible to the author.		
Assumptions:	N/A		

2.7 Settings Management

2.7.1 View settings

UC ID and Name:	UC-48 View settings		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner, Guest	Secondary Actors:	
Trigger:	1. User clicks on the “Settings” button.		
Description:	Allow users to view the available system settings.		
Preconditions:	1. User is currently using the website 2. User has signed in to the system.		
Post-conditions:	1. The system displays all the settings options to the user.		
Normal Flow:	1. User goes to the website 2. The system fetches data from the server. 3. User clicks on the “Settings” button. 4. The system redirects the user to the settings page. 5. The system displays all the settings options to the user.		
Alternative Flows:	N/A		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	N/A		
Other Information:	The settings data is fetched and stored when a user accesses the website for the first time.		
Assumptions:	N/A		

2.7.2 Update system appearance

UC ID and Name:	UC- 49 Update system appearance		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner, guest, Administrator	Secondary Actors:	

Trigger:	1. User change a setting options
Description:	Allow users to customize how the system looks to them
Preconditions:	N/A
Post-conditions:	1. The system displays a message notifying the user.
Normal Flow:	<p>A. User changes the system language.</p> <ol style="list-style-type: none"> 1. User clicks on the language dropdown. 2. User chooses one of the options “English” or “Tiếng Việt”. 3. The system updates the appearance based on the option changes. 4. The user clicks on the “Save changes” button. 5. The system sends a request to the server. 6. The system displays a message notifying the user.
Alternative Flows:	<p>A. User changes the system theme.</p> <ol style="list-style-type: none"> 1. User clicks on the language dropdown. 2. User chooses one of the options “System”, “Light” or “Dark”. 3. (Continue from step 3 of the normal flow.) <p>B. Learner hides some components or their profile appearance.</p> <ol style="list-style-type: none"> 1. User clicks on an option switch. 2. (Continue from step 3 of the normal flow.)
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	Medium
Frequency of Use:	High
Business Rules:	N/A
Other Information:	<ol style="list-style-type: none"> 1. Guests and the Administrator can only change the language of the system when using the website. 2. Language changes by guests or the Administrator are stored locally on their machine, and they will have to change it again on other machines.
Assumptions:	N/A

2.7.3 Set rest reminder

UC ID and Name:	UC-50 Set rest reminder		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner, Guest	Secondary Actors:	
Trigger:	1. The user sets a specific time for the reminder.		
Description:	The system allows users to set a specific time for rest reminders. When the set time is reached, the user will automatically receive a notification prompting them to take a break.		
Preconditions:	1. User is currently using the website 2. User has signed in to the system.		
Post-conditions:	1. The system displays a message notifying the user		
Normal Flow:	1. User goes to the settings page. 2. User sets a specific timer. 3. User clicks on the “Save changes” button. 4. The system sends a request to the server. 5. The system displays a message notifying the user.		
Alternative Flows:	N/A		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:	N/A		
Other Information:	N/A		
Assumptions:	User has the needs to be reminded about resting time.		

2.7.4 Reset settings

UC ID and Name:	UC-51 Reset settings		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner, Guest	Secondary Actors:	
Trigger:	1. User clicks on the “Reset” button.		
Description:	Allow users to reset their custom settings back to default options.		
Preconditions:	1. User is currently using the website 2. User has signed in to the system.		
Post-conditions:	1. The system displays a message notifying the user.		
Normal Flow:	1. User clicks on the “Reset” button. 2. The system displays a modal asking the user to confirm resetting. 3. User clicks on the “Reset” button. 4. The system sends a request to the server. 5. The system displays a message notifying the user.		
Alternative Flows:	N/A		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:	N/A		
Other Information:	N/A		
Assumptions:	User accidentally updated wrong settings and has the need to reset them to default.		

2.8 Learner Management

2.8.1 View learners

UC ID and Name:	UC-62 View learners		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner, guest, Administrator	Secondary Actors:	

Trigger:	1. User requests to view learners information (search results page, group detail page)
Description:	Allow users to view learners who have valid accounts inside the system.
Preconditions:	N/A
Post-conditions:	1. The system displays a list of learners to the user
Normal Flow:	<p>1. User requests to view learners information.</p> <p>2. The system fetches data from the server.</p> <p>3. The system displays a list of users to the user.</p>
Alternative Flows:	<p>A. User filters search result by author's username</p> <p>1. User input keyword into the search bar.</p> <p>2. The system redirects the user to the search result page.</p> <p>3. User clicks on the filter by author option.</p> <p>4. The system fetches the data from the server.</p> <p>5. The system displays a list of found users to the user.</p> <p>B. User is inside group detail page</p> <p>1. The system fetches the data from the server.</p> <p>2. The system displays a list of group members to the user.</p>
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	BR-08, BR-13, BR-23
Other Information:	The Administrator must go to the learners page to view learners' information.
Assumptions:	N/A

2.8.2 View learner profile

UC ID and Name:	UC-63 View learner profile		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner, guest, Administrator	Secondary Actors:	
Trigger:	1. User goes to the learner profile page.		

Description:	Allow users to view other learner profile data. Including following status, personal information, created question sets or participating public groups.
Preconditions:	1. The learner account is still active in the system.
Post-conditions:	1. The system displays data related to the learner to the user.
Normal Flow:	<ol style="list-style-type: none"> 1. User goes to the learner profile page. 2. The system fetches the data from the server. 3. The system displays data related to the learner to the user.
Alternative Flows:	<p>A. Guest and Learner is currently inside the question set detail page.</p> <ol style="list-style-type: none"> 1. User clicks on the question set author row. 2. The system redirects the user to the learner profile page. 3. (Continues from step 2 of the normal flow) <p>B. User view a non-existent learner profile.</p> <ol style="list-style-type: none"> 1. User inputs the URL into the browser search bar. 2. The system redirects the user to the not found page.
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	BR-8, BR-13, BR-14, BR-18, BR-19, BR-23
Other Information:	N/A
Assumptions:	N/A

2.8.3 Update learner accessibilities

UC ID and Name:	UC-52 Update learner accessibilities		
Created By:	Mai Nhat Dang	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User click on the “Block”/“Unblock” button inside learner table row		
Description:	Allow the Administrator to block or unblock a learner account.		
Preconditions:	1. User has signed in to the system.		

Post-conditions:	1. The system displays the status of the learner inside the table row.
Normal Flow:	<p>1. User clicks on the “Block” button on the learner table row.</p> <p>2. The system sends the request to the server.</p> <p>3. The system updates the “Block” button to “Unblock” button to the user.</p>
Alternative Flows:	<p>A. The Administrator unlock the learner</p> <p>1. User clicks on the “Block” button on the learner table row.</p> <p>2. The system sends the request to the server.</p> <p>3. The system updates the “Block” button to “Unblock” button to the user.</p> <p>B. The Administrator blocks the user while resolving the report.</p> <p>1. User clicks on the “Resolve” dropdown button.</p> <p>2. User clicks on “Block user” from the dropdown.</p> <p>3. The system displays a confirmed modal.</p> <p>4. User clicks on the “Resolve” button.</p> <p>5. The system sends the request to the server.</p> <p>6. The system displays a message to notify the user the report has been resolved.</p>
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-08, BR-23
Other Information:	N/A
Assumptions:	The Administrator acknowledge the reason why a learner should be block/unblock

2.9 Report Management

2.9.1 Create reports

UC ID and Name:	UC-20 Create report		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the “Report” button.		

Description:	Allow users to report the inappropriate question set, comments or groups
Preconditions:	1. The reported target is still active.
Post-conditions:	1. The system displays a message to notify the user.
Normal Flow:	<ol style="list-style-type: none"> 1. User clicks on the “Report” button 2. The system displays a report modal. 3. User fills in the description of the report. 4. User clicks on the “Report” button. 5. The system sends the data to the server. 6. The system displays a message to notify the user.
Alternative Flows:	N/A
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	Medium
Frequency of Use:	Medium
Business Rules:	N/A
Other Information:	N/A
Assumptions:	N/A

2.9.2 View reports

UC ID and Name:	UC-64 View reports		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User goes to the reports page.		
Description:	Allow the Administrator to view user reports sent by learners.		
Preconditions:	1. The user has signed in to the system.		
Post-conditions:	1. The system displays a table of created reports to the user.		
Normal Flow:	<ol style="list-style-type: none"> 1. User goes to the reports page. 2. The system fetches the data from the server. 3. The system displays a table of created reports to the user. 		

Alternative Flows:	N/A
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-08
Other Information:	N/A
Assumptions:	N/A

2.9.3 View report detail

UC ID and Name:	UC-65 View report detail		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User goes to the report detail page.		
Description:	Allow the Administrator to view the report detail sent by learners		
Preconditions:	1. User has signed in to the system.		
Post-conditions:	1. The system displays detailed data related to a report to the user		
Normal Flow:	1. User goes to the report detail page. 2. The system fetches the data from the server. 3. The system displays detailed data related to a report to the user		
Alternative Flows:	N/A		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-08		
Other Information:	N/A		

Assumptions:	N/A
--------------	-----

2.9.4 Update report statuses

UC ID and Name:	UC-55 Update report statuses		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User clicks on the “Resolve” button dropdown.		
Description:	Allow the Administrator to resolve the reports sent by learners.		
Preconditions:	1. The user has signed in to the system.		
Post-conditions:	1. The system displays a message to notify the user the report has been resolved.		
Normal Flow:	1. User clicks on the “Resolve” button dropdown. 2. User clicks on an option from the dropdown. 3. The system displays a confirmed modal. 4. User clicks on the “Resolve” button. 5. The system sends the request to the server. 6. The system displays a message to notify the user the report has been resolved.		
Alternative Flows:	N/A		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-08, BR-26		
Other Information:	N/A		
Assumptions:	N/A		

2.10 Field Management

2.10.1 View fields

UC ID and Name:	UC-69 View fields		
Created By:	Huynh Nguyen Gia Bao	Date Created:	03/02/2025
Primary Actor:	Learner, guest, Administrator	Secondary Actors:	
Trigger:	1. User goes to the Explore page.		
Description:	Allow the users to view the current fields inside the system.		
Preconditions:	N/A		
Post-conditions:	1. The system displays a list of created fields to the user		
Normal Flow:	1. User goes to the explore page. 2. The system fetches the data from the server. 3. The system displays a list of created fields to the user.		
Alternative Flows:	A. The Administrator view created fields 1. User goes to the fields page. 2. The system fetches the data from the server. 3. The system displays a table of created fields to the user.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-08		
Other Information:	The Administrator can not goes to the explore page to view fields		
Assumptions:	N/A		

2.10.2 View field detail

UC ID and Name:	UC-70 View field detail		
Created By:	Huynh Nguyen Gia Bao	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	

Trigger:	1. User goes to the field detail page.
Description:	Allow the Administrator to view the details of created fields.
Preconditions:	1. User has signed in to the system.
Post-conditions:	1. The system displays detailed data related to a field to the user
Normal Flow:	1. User goes to the field detail page. 2. The system fetches the data from the server. 3. The system displays detailed data related to a field to the user
Alternative Flows:	N/A
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	BR-08
Other Information:	N/A
Assumptions:	N/A

2.10.3 Create fields

UC ID and Name:	UC-71 Create fields		
Created By:	Huynh Nguyen Gia Bao	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User clicks on the “Create” button from the fields page.		
Description:	Allow the Administrator to create new fields for the system, these will be used for learners to easily find related subjects		
Preconditions:	1. User has signed in to the system.		
Post-conditions:	1. The system updates the fields list to the user.		
Normal Flow:	1. User clicks on the “Create” button from the fields page. 2. The system redirects the user to the create field page. 3. User enters all the necessary fields. 4. User clicks on the “Create” button.		

	<p>5. The system sends the data to the server.</p> <p>6. The system redirects the user to the fields page.</p> <p>7. The system updates the fields list to the user.</p>
Alternative Flows:	<p>A. User create new field with empty required field</p> <ol style="list-style-type: none"> 1. (Start from step 1 to step 2 of the normal flow) 2. User leaves required fields as empty 3. User clicks on the “Create” button 4. The system displays missing data message to the user <p>B. User create a field that already existed</p> <ol style="list-style-type: none"> 1. (Start from step 1 to step 5 of the normal flow) 2. The system displays duplicated data message to the user.
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	N/A
Business Rules:	BR-08
Other Information:	N/A
Assumptions:	N/A

2.10.4 Update fields

UC ID and Name:	UC-72 Update fields		
Created By:	Huynh Nguyen Gia Bao	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User clicks on the “Edit” button on the fields table row.		
Description:	Allow the Administrator to update the created fields.		
Preconditions:	1. User has signed in to the system.		
Post-conditions:	1. The system updates the data to the user.		
Normal Flow:	1. User clicks on the “Edit” button on the fields table row.. 2. The system redirects the user to the update field page.		

	<p>3. The system fetches the related data from the server.</p> <p>4. The system displays the data to the user.</p> <p>5. User changes their desired data.</p> <p>6. User clicks the “Update” button.</p> <p>7. The system sends the modified data to the server.</p> <p>8. The system redirects the user to the fields page.</p> <p>9. The system updates the data to the user.</p>
Alternative Flows:	<p>A. User update new field with empty required field</p> <p>1. (Start from step 1 to step 5 of the normal flow)</p> <p>2. User leaves required fields as empty</p> <p>3. User clicks on the “Update” button</p> <p>4. The system displays missing data message to the user</p> <p>B. User create a field that already existed</p> <p>1. (Start from step 1 to step 7 of the normal flow)</p> <p>2. The system displays duplicated data message to the user.</p>
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-08
Other Information:	N/A
Assumptions	N/A

2.10.5 Delete fields

UC ID and Name:	UC-73 Delete fields		
Created By:	Huynh Nguyen Gia Bao	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User clicks on the delete button on the fields table row		
Description:	Allow the Administrator to delete the created fields		

Preconditions:	1. User has signed in to the system.
Post-conditions:	1. The system displays the updated fields table to the user.
Normal Flow:	<ol style="list-style-type: none"> 1. User clicks on the delete button on the fields table row. 2. The system shows the modal to confirm user action. 3. User clicks on the “Delete” button. 4. The system sends the request to the server. 5. The system displayed an updated fields table to the user.
Alternative Flows:	N/A
Exceptions:	<ol style="list-style-type: none"> 1. User clicks the “Cancel” button. The system cancels the action and hides the delete modal.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-08, BR-28
Other Information:	N/A
Assumptions:	N/A

2.11 Subject Management

2.11.1 View subjects

UC ID and Name:	UC-58 View subjects		
Created By:	Huynh Nguyen Gia Bao	Date Created:	03/02/2025
Primary Actor:	Learner, guest, Administrator	Secondary Actors:	
Trigger:	1. User goes to the Explore page.		
Description:	Allow the Administrator to view the current subjects on the system.		
Preconditions:	N/A		
Post-conditions:	1. The system displays a list of created subjects to the user.		
Normal Flow:	<ol style="list-style-type: none"> 1. User goes to the subjects page. 2. The system fetches the data from the server. 3. The system displays a list of created subjects to the user. 		

Alternative Flows:	A. The Administrator view created subjects 1. User goes to the subjects page. 2. The system fetches the data from the server. 3. The system displays a table of created subjects to the user.
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	BR-08
Other Information:	The Administrator can not goes to the explore page to view subjects
Assumptions:	N/A

2.11.2 View subject detail

UC ID and Name:	UC-74 View subject detail		
Created By:	Huynh Nguyen Gia Bao	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User goes to the subject detail page.		
Description:	Allow the Administrator to view the details of a created subject.		
Preconditions:	1. User has signed in to the system.		
Post-conditions:	1. The system displays detailed data related to a subject to the user		
Normal Flow:	1. User goes to the subject detail page. 2. The system fetches the data from the server. 3. The system displays detailed data related to a subject to the user		
Alternative Flows:	N/A		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-08		

Other Information:	N/A
Assumptions:	N/A

2.11.3 Create subjects

UC ID and Name:	UC-59 Create subjects		
Created By:	Huynh Nguyen Gia Bao	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User clicks on the “Create” button from the subjects page.		
Description:	Allow the Administrator to create new subjects for the system, these will be used for learners to categories their question sets.		
Preconditions:	1. User has signed in to the system.		
Post-conditions:	1. The system updates the subjects list to the user.		
Normal Flow:	1. User clicks on the “Create” button from the subjects page. 2. The system redirects the user to the create subjects page. 3. User enters all the necessary fields. 4. User clicks on the “Create” button. 5. The system sends the data to the server. 6. The system redirects the user to the subjects page. 7. The system updates the fields list to the user.		
Alternative Flows:	A. User create new subject with empty required field 1. (Start from step 1 to step 2 of the normal flow) 2. User leaves required fields as empty 3. User clicks on the “Create” button 4. The system displays missing data message to the user B. User create a subject that already existed 1. (Start from step 1 to step 5 of the normal flow) 2. The system displays duplicated data message to the user.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		

Frequency of Use:	N/A
Business Rules:	BR-08
Other Information:	N/A
Assumptions:	N/A

2.11.4 Update subjects

UC ID and Name:	UC-60 Update subjects		
Created By:	Huynh Nguyen Gia Bao	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User clicks on the “Edit” button on the subjects table row.		
Description:	Allow the Administrator to update the created subjects.		
Preconditions:	1. User has signed in to the system.		
Post-conditions:	1. The system updates the data to the user.		
Normal Flow:	1. User clicks on the “Edit” button on the subjects table row.. 2. The system redirects the user to the update subject page. 3. The system fetches the related data from the server. 4. The system displays the data to the user. 5. User changes their desired data. 6. User clicks the “Update” button. 7. The system sends the modified data to the server. 7. The system redirects the user to the subjects page. 8. The system updates the data to the user.		
Alternative Flows:	A. User update new field with empty required field 1. (Start from step 1 to step 5 of the normal flow) 2. User leaves required fields as empty 3. User clicks on the “Update” button 4. The system displays missing data message to the user B. User create a field that already existed 1. (Start from step 1 to step 7 of the normal flow)		

	2. The system displays duplicated data message to the user.
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-08
Other Information:	N/A
Assumptions:	N/A

2.11.5 Delete subjects

UC ID and Name:	UC-61 Delete subjects		
Created By:	Huynh Nguyen Gia Bao	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User clicks on the delete button on the subject table row		
Description:	Allow the Administrator to delete the created subjects		
Preconditions:	1. User has signed in to the system.		
Post-conditions:	1. The system displays the updated subjects table to the user.		
Normal Flow:	1. User clicks on the delete button on the subject table row. 2. The system shows the modal to confirm user action. 3. User clicks on the “Delete” button. 4. The system sends the request to the server. 5. The system displayed an updated subjects table to the user.		
Alternative Flows:	N/A		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	Medium		

Business Rules:	BR-08, BR-28
Other Information:	N/A
Assumptions:	N/A

2.12 Following Management

2.12.1 Follow account

UC ID and Name:	UC-08 Follow account		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the “Follow” button		
Description:	Allow users to follow other user accounts.		
Preconditions:	1. User has signed in to the system.		
Post-conditions:	1. The system updates to show that the user has been followed		
Normal Flow:	1. User clicks on the “Follow” button. 2. The system requests the server to update the status. 3. The system changes the “Follow” button title to “Following”.		
Alternative Flows:	A. User is currently inside the group detail page. 1. User goes to the group detail page. 2. User clicks on the “Member” tabs. 3. (Continue from step 1 of Normal flow) B. User is currently inside the question set detail page. 1. User goes to the question set detail page. 2. (Continue from step 1 of Normal flow) C. User unfollows a learner. 1. User clicks on the “Following” button. 2. The system requests the server to update the status. 3. The system changes the “Following” button title to “Follow”.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		

Frequency of Use:	High
Business Rules:	BR-02
Other Information:	N/A
Assumptions:	N/A

2.12.2 View follow detail

UC ID and Name:	UC-09 View follow detail		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner, Guest, Administrator	Secondary Actors:	
Trigger:	1. User click on the followings or followers number		
Description:	Allow users to view the following and follower details of the learner.		
Preconditions:	1. User is currently inside the learner detail page.		
Post-conditions:	1. The system displays the follow detail modal		
Normal Flow:	1. User clicks on the followings or followers number. 2. The system requests the data from the server. 3. The system displays the follow detail modal		
Alternative Flows:	A. Administrator views a learner's follow detail. 1. User goes to the learners page. 2. User clicks on a learner table row. 3. The system redirected the user to the learner detail page. 4. The system fetches the data from the server. 5. The system displays the data to the user 6. User scroll down to the follow section.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-08		
Other Information:	N/A		

Assumptions:	N/A
--------------	-----

2.13 Question Set Progress Management

2.13.1 Mark questions

UC ID and Name:	UC-17 Mark questions		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the star icon from the question card inside the question set detail page.		
Description:	Allow users to mark questions inside a question set, users later can filter out questions based on these marked questions.		
Preconditions:	1. User has signed in to the system. 2. The current question is still active. 3. User is currently inside the question set detail page.		
Post-conditions:	1. The system updates the look of the star icon of the current question card.		
Normal Flow:	1. User scrolls to the question they want to mark. 2. User clicks on the star icon. 3. The system sends the request to the server. 4. The system updates the look of the star icon of the current question card.		
Alternative Flows:	A. User marked a question during learn mode 1. User clicks on the “Learn” button. 2. (Continue from step 2 of the Normal flow)		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	N/A		
Other Information:	Learners can click on the star icon of the marked question to unmark the question.		
Assumptions:	N/A		

2.13.2 Filter questions

UC ID and Name:	UC-21 Filter questions		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner, Guest	Secondary Actors:	
Trigger:	1. User inputs keywords into the search bar inside the question set detail page.		
Description:	Allow users to filter questions inside a question set using keywords or with available filter options.		
Preconditions:	1. User is currently inside the question set detail page.		
Post-conditions:	1. The system displays a list of questions that match the user filter.		
Normal Flow:	1. User inputs keywords into the search bar inside the question set detail page. 2. The system filters data. 3. The system displays a list of questions that match the user filter.		
Alternative Flows:	A. Learners filter questions based on the learning progress of the question. 1. User clicks on an option inside the filter dropdown. 2. (continues from step 2 of the Normal flow). A. Learners filter for non-existent question. 1. (Start from step 1 to step 2 of the normal flow) 2. The system displays an empty message to the user.		
Exceptions:	N/A		
Priority:	High		
Frequency of Use:	High		
Business Rules:	N/A		
Other Information:	N/A		
Assumptions:	N/A		

2.14 Comment Management

2.14.1 View comments

UC ID and Name:	UC-25 View comments		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Learner, guest, Administrator	Secondary Actors:	
Trigger:	1. User goes to the question set detail page.		
Description:	Allow users to view comments created by other learners inside a question set.		
Preconditions:	1. The current question set is still active		
Post-conditions:	1. The system displays a section of comments to the user.		
Normal Flow:	1. User goes to the question set detail page. 2. User scrolls down to the comment section. 3. The system fetches the data from the server. 4. The system displays a section of comments to the user.		
Alternative Flows:	A. Users view the question comments 1. User goes to the question set detail page. 2. User scrolls to a question set question. 3. User clicks on the comment icon next to the question card. 4. The system fetches the data from the server. 5. The system displays a section of question comment inside comment modal.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-08, BR-12, BR-13		
Other Information:	N/A		
Assumptions:	N/A		

2.14.2 Filter comments

UC ID and Name:	UC-22 Filter comments		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Learner, Guest	Secondary Actors:	
Trigger:	1. User chooses a filter option from the comment section.		
Description:	Allow users to filter comments base on date created		
Preconditions:	1. User is currently inside the question set detail page. 2. The current question set is still active.		
Post-conditions:	1. The system displays a section of related comments to the user.		
Normal Flow:	1. User chooses a filter option from the comment section 2. The system fetches the data from the server. 3. The system displays a section of related comments to the user.		
Alternative Flows:	A. User filters the comments in question comments 1. User clicks on the comment icon next to the question card. 2. User chooses a filter option inside the comment modal. 3. The system fetches the data from the server. 4. The system displays a section of question comment inside comment modal.		
Exceptions:	N/A		
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:	BR-13		
Other Information:	N/A		
Assumptions:	N/A		

2.14.3 Create comments

UC ID and Name:	UC-26 Create comments		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	

Trigger:	1. User inputs a comment. 2. User clicks on the “Comment” button.
Description:	Allow users to create comments inside a question set.
Preconditions:	1. User is currently inside the question set detail page. 2. User has signed in to the system.
Post-conditions:	1. The system displays the comment at the top of the comment section.
Normal Flow:	1. User clicks on the comment input. 2. User inputs a comment. 3. User clicks on the “Comment” button. 4. The system sends the data to the server. 5. The system displays the comment at the top of the comment section.
Alternative Flows:	<p>A. User creates a question comment.</p> <ol style="list-style-type: none"> 1. User clicks on the comment icon next to the question card. 2. User inputs a comment. 3. User clicks on the “Comment” button. 4. The system sends the date to the server. 5. The system displays the comment at the top of the comment modal. <p>B. User creates an empty comment.</p> <ol style="list-style-type: none"> 1. User clicks on the comment input. 2. User leave the comment as empty. 2. The system disable the comment button.
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	N/A
Other Information:	N/A
Assumptions:	N/A

2.14.4 Mark question comments

UC ID and Name:	UC-23 Mark question comments		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User hovers and clicks on the check mark icon inside a question comment		
Description:	Allow the author of the question set to mark a question comment, which means the question comment is approved by the author.		
Preconditions:	1. The user has signed in to the system. 2. Current user is the author of the question set. 3. The question set is still active.		
Post-conditions:	1. The system displays a checkmark next to the marked question comment.		
Normal Flow:	1. User hovers and clicks on the check mark icon inside a question comment. 2. The system sends the request to the server. 3. The system displays a checkmark next to the marked question comment. 4. The system displays a message to notify the user the comment has been marked.		
Alternative Flows:	N/A		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	Medium		
Business Rules:	BR-04, BR-12, BR-14		
Other Information:	If the author marks another question comment inside the question that already has a mark comment, the new one will replace the old one.		
Assumptions:	N/A		

2.14.5 Update comments

UC ID and Name:	UC-66 Update comments		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Learners	Secondary Actors:	
Trigger:	1. User clicks on the edit option inside the action dropdown of the comments. 2. User edits the comment. 3. User clicks on the “Edit” button.		
Description:	Allow users to edit their created comments.		
Preconditions:	1. User has signed in to the system. 2. User is currently inside the question set detail page. 3. The current question set is active. 4. User is the author of the current comment.		
Post-conditions:	1. The system displays the updated comment to the user.		
Normal Flow:	1. User clicks on the edit option inside the action dropdown of the comments. 2. User edits the comment. 3. User clicks on the “Edit” button. 4. The system sends the data to the server. 5. The system displays the updated comment to the user.		
Alternative Flows:	A. User updates a question comment. 1. User clicks on the comment icon next to the question card. 2. (continue from step 1 to step 4 of the normal flow) 5. The system displays the updated comment inside the comment modal. B. User updates an empty comment. 1. User clicks on the edit option inside the action dropdown of the comments. 2. User leave the comment as empty. 3. The system disable the comment button.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	Medium		
Business Rules:	BR-17		

Other Information:	N/A
Assumptions:	N/A

2.14.6 Delete comments

UC ID and Name:	UC-28 Delete comments		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the delete option inside the action dropdown of the comments.		
Description:	Allow users to delete their created comment		
Preconditions:	1. User has signed in to the system. 2. User is currently inside the question set detail page. 3. The current question set is active. 4. User is the author of the current comment.		
Post-conditions:	1. The system removes the comment from the comment section.		
Normal Flow:	1. User clicks on the delete option inside the action dropdown of the comments. 2. The system displays the confirm modal. 3. User clicks on the “Delete” button. 4. The system sends the data to the server. 5. The system removes the comment from the comment section.		
Alternative Flows:	A. User deletes a question comment. 1. User clicks on the comment icon next to the question card. 2. (continue from step 1 to step 4 of the normal flow) 3. The system removes the comment inside the comment modal.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	Medium		
Business Rules:	BR-17		
Other Information:	N/A		
Assumptions:	N/A		

2.14.7 Reply comments

UC ID and Name:	UC-27 Reply comments		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the “Reply” button under a comment. 2. User enters the reply comment. 3. User clicks on the “Reply” button.		
Description:	Allow users to create replies to a comment inside a question set.		
Preconditions:	1. User has signed in to the system. 2. User is currently inside the question set detail page. 3. The current question set is active.		
Post-conditions:	1. The system displays the replied comment under the initial comment.		
Normal Flow:	1. User clicks on the “Reply” button under a comment. 2. User enters the reply comment. 3. User clicks on the “Reply” button. 4. The system sends the data to the server. 5. The system displays the replied comment under the initial comment.		
Alternative Flows:	A. User replies to a question comment. 1. User clicks on the comment icon next to the question card. 2. (continue from step 1 of the normal flow)		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	Medium		
Business Rules:	BR-17		
Other Information:	N/A		
Assumptions:	N/A		

2.14.8 Upvote comments

UC ID and Name:	UC-29 Upvote comments		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the thumb up icon under a comment.		
Description:	Allow user to upvote a comment inside question set		
Preconditions:	1. User has signed in to the system. 2. User is currently inside the question set detail page. 3. The current question set is active.		
Post-conditions:	The system displays the updated number of upvotes to the user.		
Normal Flow:	1. User clicks on the thumb up icon under a comment. 2. The system sends the request to the server. 3. The system displays the updated number of upvotes to the user.		
Alternative Flows:	A. Users upvote a question comment. 1. User clicks on the comment icon next to the question card. 2. (continue from step 1 of the normal flow)		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	N/A		
Business Rules:	BR-17		
Other Information:	N/A		
Assumptions:	N/A		

2.15 Learn Mode Interaction

2.15.1 Use learn mode

UC ID and Name:	UC-31 Use learn mode		
Created By:	Huynh Nguyen Gia Bao	Date Created:	03/02/2025
Primary Actor:	Learner, guest	Secondary Actors:	

Trigger:	1. User clicks on the “Learn” button inside the question set detail page.
Description:	Allow users to learn a question set and have the system tracking their progress.
Preconditions:	1. The current question set is still active.
Post-conditions:	1. The system displays the questions as flash cards for the user.
Normal Flow:	<p>1. User clicks on the “Learn” button inside the question set detail page.</p> <p>2. The system redirects the user to the learn page.</p> <p>3. The system fetches the data from the server.</p> <p>4. The system displays the questions as flash cards for the user.</p>
Alternative Flows:	<p>A. User customizes learning options.</p> <p>1. (start from step 1 to step 3 of the normal flow).</p> <p>2. User clicks on the “Learn options” button.</p> <p>3. The system displays the option modals to the user.</p> <p>4. User customizes the option and clicks the “Start learning” button.</p> <p>5. The system displays the questions as flash cards for the user.</p>
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	N/A
Business Rules:	BR-22, BR-27, BR-29
Other Information:	N/A
Assumptions:	N/A

2.15.2 Use matching mode

UC ID and Name:	UC-67 Use matching mode		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner, guest	Secondary Actors:	
Trigger:	1. User clicks on the “Matching” button inside the question set detail page.		
Description:	Allow users to learn a question set by playing matching game,		
Preconditions:	1. The current question set is still active.		

Post-conditions:	1. The system displays the questions and answers as separate cards for the user.
Normal Flow:	<ol style="list-style-type: none"> 1. User clicks on the “Matching” button inside the question set detail page. 2. The system redirects the user to the matching page. 3. The system fetches the data from the server. 4. The system displays the questions and answers as separate cards for the user.
Alternative Flows:	<p>A. User is currently inside the learn page.</p> <ol style="list-style-type: none"> 1. User chooses the “Matching” option inside the learn header dropdown. 2. (continue from step 2 of the normal flow).
Exceptions:	<ol style="list-style-type: none"> 1. User cancels the matching attempt. The system displays a pop up warning the user. If the user persists, the system cancels the action and returns the user back to the question set detail page. 2. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	Medium
Business Rules:	BR-29
Other Information:	N/A
Assumptions:	N/A

2.16 Quiz Mode Interaction

2.16.1 Use quiz mode

UC ID and Name:	UC-32 Use quiz mode		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner, guest	Secondary Actors:	
Trigger:	1. User clicks on the “Quiz” button inside the question set detail page.		
Description:	Allow users to learn a question set by doing quizzes, the system will then calculate the final score and feedback for the user.		
Preconditions:	1. The current question set is still active		
Post-conditions:	1. The system displays the total score to the users.		
Normal Flow:	<ol style="list-style-type: none"> 1. User clicks on the “Quiz” button inside the question set detail page. 2. The system redirects the user to the quiz page. 		

	<p>3. The system displays the quiz option modal.</p> <p>4. User customizes the option and clicks the start button.</p> <p>5. The system fetches the data from the server.</p> <p>6. The system displays the question as quizzes.</p> <p>7. User chooses/ inputs the answer and clicks the “Submit” button.</p> <p>8. The system sends the data to the server.</p> <p>9. The system displays the total score to the users.</p>
Alternative Flows:	<p>A. User is currently inside the learn page.</p> <p>1. User chooses the “Quiz” option inside the learn header dropdown.</p> <p>2. (continue from step 2 of the normal flow).</p> <p>B. Learner chooses a quiz inside quiz statistics.</p> <p>1. User chooses the “Quiz” option inside question set statistics.</p> <p>2. (continue from step 2 of the normal flow).</p>
Exceptions:	<p>1. User cancels the quiz attempt. The system displays a pop up warning the user. If the user persists, the system cancels the action and returns the user back to the question set detail page.</p> <p>2. User loses network connection. The system displays error message.</p>
Priority:	High
Frequency of Use:	High
Business Rules:	BR-29
Other Information:	N/A
Assumptions:	N/A

2.16.2 Revisit quiz attempts

UC ID and Name:	UC-33 Revisit quiz attempts		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the “Revisit” button inside quiz result card		
Description:	Allow users to revisit their previous quiz attempts, which displays what users have answered and total score.		
Preconditions:	<p>1. User has signed in to the system.</p> <p>2. The current question set is still active.</p> <p>2. User has done a quiz attempt.</p>		

Post-conditions:	1. The system displays the details of previous quiz attempts to the user.
Normal Flow:	<ol style="list-style-type: none"> 1. User clicks on the “Revisit” button inside the quiz result card. 2. The system redirects the user to the quiz attempt page. 3. The system fetches the data from the server. 4. The system displays the details of previous quiz attempts to the user.
Alternative Flows:	N/A
Exceptions:	1. User loses network connection. The system displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	BR-30
Other Information:	N/A
Assumptions:	The attempt the user wants to revisit is still one of the 10 latest quiz attempts.

2.16.3 View quiz statistics

UC ID and Name:	UC-34 View quiz statistics		
Created By:	Tran Le Hoang Long	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	<ol style="list-style-type: none"> 1. User goes to the question set detail page. 		
Description:	Allow users to view the overall statistics of their question set progress and previous quiz attempts.		
Preconditions:	<ol style="list-style-type: none"> 1. User has signed in to the system. 2. The current question set is still active 3. User has done at least one quiz. 		
Post-conditions:	1. The system displays the statistics data to the user.		
Normal Flow:	<ol style="list-style-type: none"> 1. User goes to the question set detail page. 2. The system fetches data from the server. 3. The system displays the statistics data to the user. 		
Alternative Flows:	N/A		
Exceptions:	1. User loses network connection. The system displays error message.		

Priority:	High
Frequency of Use:	High
Business Rules:	BR-30
Other Information:	N/A
Assumptions:	N/A

2.17 Statistics Interaction

2.17.1 View statistics

UC ID and Name:	UC-62 View statistics		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. User goes to the Dashboard page.		
Description:	Allow the Administrator to view the system's overall statistics.		
Preconditions:	1. User has signed in to the system.		
Post-conditions:	1. The system displays related data to the user.		
Normal Flow:	1. User goes to the Dashboard page. 2. The system fetches the data from the server. 3. The system displays related data to the user.		
Alternative Flows:	N/A		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:	BR-08		
Other Information:	N/A		
Assumptions:	N/A		

2.17.2 Filter statistics

UC ID and Name:	UC-63 Filter statistics		
Created By:	Mai Quang Vinh	Date Created:	03/02/2025
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	1. Users choose an option inside the Dashboard page.		
Description:	Allow the Administrator to filter statistics based on date range		
Preconditions:	1. The user has signed in to the system.		
Post-conditions:	1. The system displays the related data to the user.		
Normal Flow:	1. Users choose an option inside the Dashboard page. 2. The system fetches the data from the server. 3. The system displays the related data to the user.		
Alternative Flows:	N/A		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	High		
Frequency of Use:	Medium		
Business Rules:	BR-08		
Other Information:	N/A		
Assumptions:	N/A		

2.18 Additional Features

2.18.1. Use pomodoro timer

UC ID and Name:	UC-46 Use pomodoro timer		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner, guest	Secondary Actors:	
Trigger:	1. User clicks on the Pomodoro button.		
Description:	Allow users to keep track of their own working session using the Pomodoro method, including managing their own session duration.		

Preconditions:	1. The Pomodoro timer is available on the current page.
Post-conditions:	1. The timer displays a countdown of pre-configure intervals by the user.
Normal Flow:	<ol style="list-style-type: none"> 1. User clicks on the Pomodoro button. 2. User clicks on the Start button inside the Pomodoro overlay. 3. The timer starts counting down. 4. The timer displays a sound to notify the timer has reached 0. 5. The timer change the Start button to Pause button
Alternative Flows:	<p>A. The user modify the timer before usage</p> <ol style="list-style-type: none"> 1. User clicks on the Pomodoro button. 2. User clicks on the setting button inside the Pomodoro overlay. 3. User modifies the timer settings inside the settings modal. 4. User closes the settings modal. 5. (continue from step 2 of the normal flow). <p>B. The user switch to rest mode</p> <ol style="list-style-type: none"> 1. User clicks on the Pomodoro button. 2. User clicks on the switch button inside the Pomodoro overlay 3. The timer switches to rest mode. 4. (continue from step 2 of the normal flow). <p>C. The user set duration to 0</p> <ol style="list-style-type: none"> 1. User clicks on the Pomodoro button. 2. User clicks on the setting button inside the Pomodoro overlay. 3. User set the time duration to 0. 4. The system disables the start button.
Exceptions:	N/A
Priority:	Medium
Frequency of Use:	Medium
Business Rules:	BR-25
Other Information:	N/A
Assumptions:	Users already know how the Pomodoro method works.

2.18.2. View achievements

UC ID and Name:	UC-47 View achievements		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner	Secondary Actors:	
Trigger:	1. User clicks on the “Achievements” button.		
Description:	Allow users to view available achievements in the system.		
Preconditions:	User is on the Achievement page.		
Post-conditions:	The system displays all the available achievements in the system.		
Normal Flow:	1. User clicks on the “Achievement” button. 2. The system redirects the user to the achievements page. 3. The system requests all available achievements from the server. 4. The system displays the achievements to the user.		
Alternative Flows:	A. User view achievements from the account page 1. User clicks on the profile button 2. The system redirects the user to the account page. 3. The system fetches the data from the server. 4. The system displays the achieved achievement to the user.		
Exceptions:	1. User loses network connection. The system displays error message.		
Priority:	Low		
Frequency of Use:	N/A		
Business Rules:	N/A		
Other Information:	Learners can also view the progress of each achievement.		
Assumptions:	N/A		

2.18.2. View quotes

UC ID and Name:	UC-68 View quotes		
Created By:	Do Duc Dat	Date Created:	03/02/2025
Primary Actor:	Learner, guest	Secondary Actors:	
Trigger:	1. User goes to the home page		

Description:	Allow users to view famous quotes in the system.
Preconditions:	N/A
Post-conditions:	The system displays the quote to the user.
Normal Flow:	<ol style="list-style-type: none"> 1. User goes to the home page. 2. The system gets the data from the local. 3. The system displays the quote to the user.
Alternative Flows:	<p>A. User disable quotes</p> <ol style="list-style-type: none"> 1. User clicks on the settings button. 2. The system redirects the user to the settings page. 3. User clicks on the show quotes to hide them. 4. User goes back to the home page. 5. System hides the quotes.
Exceptions:	N/A
Priority:	Low
Frequency of Use:	High
Business Rules:	N/A
Other Information:	Learners can hide this feature in the settings
Assumptions:	N/A

3. Functional Requirements

3.1 System Functional Overview

a. Screen Flow

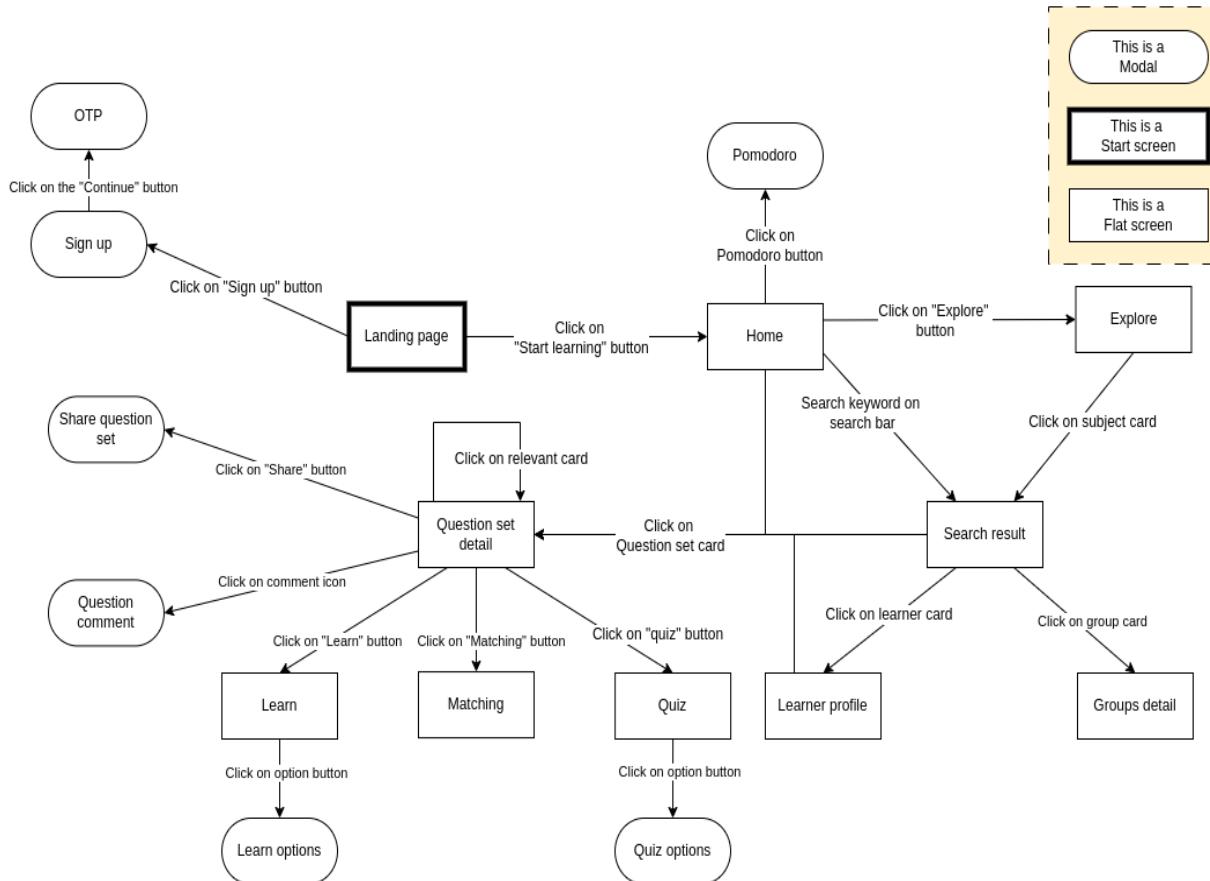


Figure 3.1.a.1.a: Guest's website Screen flow Diagram²⁶

²⁶ [guest-screenflow.drawio.png](#)

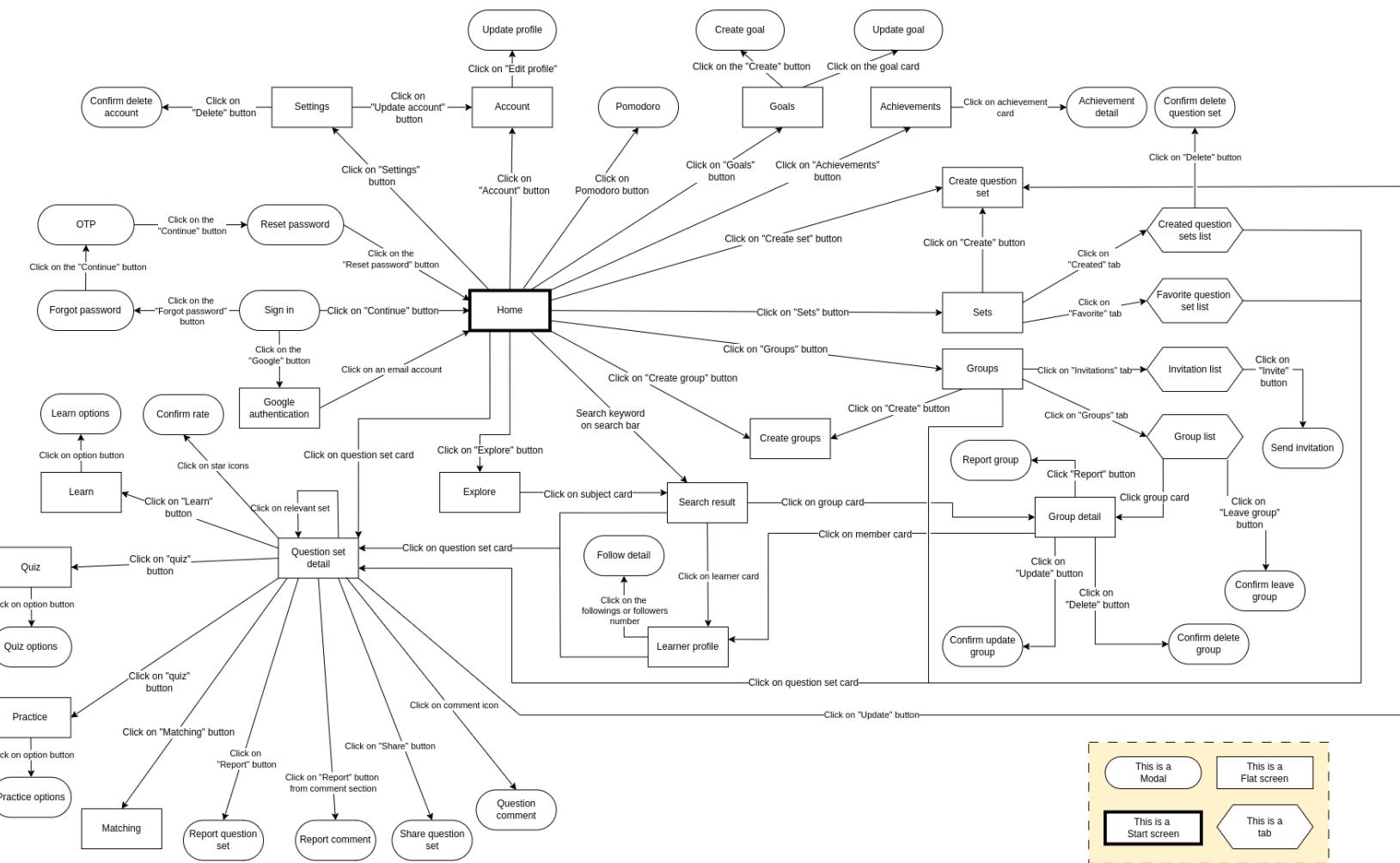


Figure 3.1.a.2.a: Learner's website Screen flow Diagram²⁷

²⁷ [learner-screenflow.drawio.png](#)

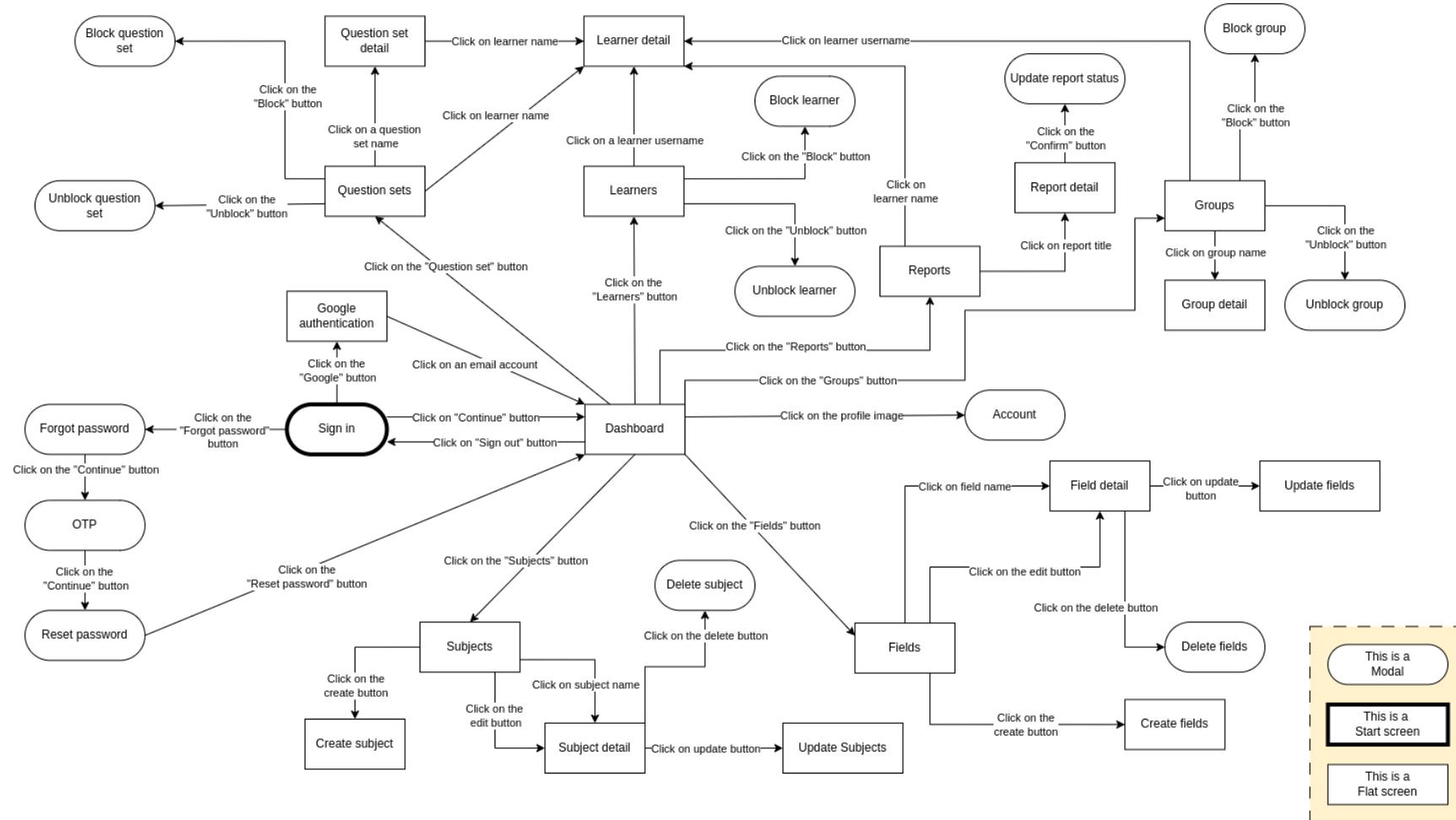


Figure 3.1.a.3: Administrator's Screen flow Diagram²⁸

²⁸ [admin-screenflow.drawio.png](#)

b. Screen Details

#	Screen	Role	Screen function	Description
01	Landing page	Guest	Landing page	Provide website introduction and the choice for authentication
02	Home	Learner, Guest	Home	Display system overall information
03	Sign in	Learner, Administrator	Sign in	Users can sign in with their credentials or with a Google account.
04	Sign up	Guest	Sign up	Users can create an account inside the system using their credentials or a Google account.
05	Forgot password	Learner, Administrator	Retrieve password	Require users to verify their account before retrieving their password.
06	OTP	Learner, Guest, Administrator	Code verification	Users receive an OTP code sent by the system to retrieve password.
07	Reset password	Learner, Administrator	Reset password	Users can reset password after successful verification
08	Google authentication	Learner, Guest, Administrator	Authenticate with Google account	Users can sign in or sign up to the system by choosing their preferred Google account.
09	Account	Learner	View profile	Users can view their profile information.
10	Update profile	Learner	Update profile	Users can update their profile information
11	Settings	Learner, Guest	View system settings	Users view available system settings
12	Confirm delete account	Learner	Delete account	When users no longer have the need to use the current account, they can delete it, the deleted account won't be able to sign in again.
13	Pomodoro	Learner, Guest	Pomodoro utilization	Users can enable and modify the Pomodoro timer inside the system to manage their working session.
14	Goals	Learner	View goals	Users view their created goal and track their progress.
15	Create goal	Learner	Create new goal	Users create a new goal to manage their own working tasks.
16	Update goal	Learner	Update a goal	Users update the progress or the status of the goal.
17	Achievements	Learner	View achievements	Users can view the achievements available inside the system and keep track of their progress.
18	Achievement detail	Learner	View achievement detail	Users view the details of the achievements, including the guide to progress and their current progress.

19	Explore	Learner, Guest	View system subjects	Users view the subjects available inside the system and filter question sets based on them.
20	Search result	Learner, Guest	Search question sets	User search with a keyword and view the question set found with the keyword. Users can advance further by customizing the filter option for the found result.
21	Question set detail	Learner, Guest	View a question set detail	Users view the detail of a questions set, including its basic information, a list of questions and the comment section. Learners can create and manage their comments. Learner can add the current question set to their favorite list
22	Learn	Learner, Guest	Learn the question set questions	Users learn the question inside the question set
23	Learn options	Learner, Guest	Customizing learning option	Users customizing learning options based on their learning references.
24	Practice	Learner	Practice the question set questions	Users practice the question inside the question set, the learners' progress will be kept track by the system.
25	Practice options	Learner	Customize practice option	Users customizing learning options based on their learning references.
26	Quiz	Learner, Guest	Take a quiz using the question set question	The system generates a quiz based on the questions of the question, learners' submissions will be kept track
27	Quiz options	Learner, Guest	Customizing quiz options	Users customizing quiz options based on their learning references.
28	Matching	Learner, Guest	Practice the question set question	Users practice the question set question by doing a matching game, this will not count into the progress.
29	Confirm rate	Learner	Confirm rating question set	Users confirm rating the question set.
30	Report question set	Learner	Report question set.	Users create a report for the Administrator about the inappropriate question set.
31	Question comment	Learner, Guest	View comments	Users view comments inside each question. Learners can create and manage their comments and get marked by the question set author.
32	Share question set	Learner, Guest	Share question set	Users share the question set as URLs to other users. Learners can share the question set to their participating groups.
33	Report comment	Learner	Report comment	Users create a report for the Administrator about the inappropriate comment.
34	Confirm update question set	Learner	Confirm update a question set	Users confirm updating a question set. This will not affect the overall progress of other learners of the update question set.

35	Learner profile	Learner, Guest	View learner profile	Users view the profile information of other learners. Learners can follow or unfollow the current learner.
36	Follow detail	Learner	View follow detail	User views the details of followings or followers of a learner.
37	Sets	Learner	View personal question sets	Users view their personal question set collections. Including favorite question sets and sets created by themselves.
38	Create / Update question set	Learner, Guest	Create new question set	Users customize and create a new question set inside the system.
39	Create / Update question set	Learner	Update a question set	Users update their created question set.
40	Created question set list	Learner	View created question sets	Users view the list of their created question sets. Users can delete a question set of themselves.
41	Confirm delete question set	Learner	Confirm delete question sets.	Users confirm deleting their created question set.
42	Favorite question set list	Learner	View favorite question sets	Users view the list of their favorite question set. Users can remove a question set from the favorite list.
43	Groups	Learner	View personal group information	Users view the overall information about their participated groups.
44	Create group	Learner	Create new group	Users customize and create a new group inside the system.
45	Group list	Learner	View groups	Users view the list of their participating groups.
46	Group detail	Learner, Guest	View group detail	Users view the details of a group. Learners can join the current groups.
47	Confirm update group	Learner	Confirm updating group	Users confirm updating their created groups.
48	Confirm delete group	Learner	Confirm delete groups	Users confirm deleting their created groups.
49	Confirm leave group	Learner	Confirm leave group	Users confirm leaving their participating groups as a group member. Group leaders have to switch their role with one of the members before leaving.
50	Invitation list	Learner	View list of group invitations	Users view the list of group invitations sent by other group members.
51	Send invitation	Learner	Send group invitations	Users as a group member can send invitations to other learners.
52	Report group	Learner	Report group	Users create a report for the Administrator about the inappropriate group.
53	Dashboard	Administrator	View system statistics	Administrator views the overall statistics of the whole system.
54	Administrator account	Administrator	Manage profile	Administrator views and updates the profile information

55	Administrator question sets	Administrator	View question sets	Administrator views the question sets created by learners.
56	Administrator question set detail	Administrator	View question set detail	Administrator views the details of a question set, including comments made by learners.
57	Block question set	Administrator	Block a question set	Administrator blocks an inappropriate question set. The question set will then not be public to learners and guests.
58	Unblock question set	Administrator	Unblock a question set	Administrator unblocks a blocked question set. The question set is then public to learners and guests.
59	Learners	Administrator	View learners	Administrator views the accounts created inside the system.
60	Learner detail	Administrator	View learner detail	Administrator views the details of a learner.
61	Block learner	Administrator	Block learner	Administrator blocks the learner from the system. The blocked learners will not be able to sign in to the system.
62	Unblock learner	Administrator	Unblock learner	Administrator unblocks the blocked learner. The learner is then able to sign in to the system.
63	Report	Administrator	View reports	Administrator views the report sent by learners and guests.
64	Report detail	Administrator	View report details	Administrator views the details of a report.
65	Update report status	Administrator	Update a report status	Administrator updates the status of a report after successfully resolves it.
66	Administrator groups	Administrator	View groups	Administrator views the groups created by learners inside the system.
67	Administrator group detail	Administrator	View group detail	Administrator views the detail of a learner group
68	Block group	Administrator	Block group	Administrator blocks a group. The blocked group will then not be public to the learners and guests.
69	Unblock group	Administrator	Unblock group	Administrator unblock a blocked group. The group is then public to the learners and guests.
70	Fields	Administrator	View fields	Administrator views the fields created inside the system.
71	Field detail	Administrator	View field detail	Administrator views the detail of a field created inside the system.
72	Create field	Administrator	Create new field	Administrator creates a new field inside the system. The learners later can use these fields when filtering for related subjects.
73	Update field	Administrator	Update field	Administrator updates a field.
74	Delete field	Administrator	Delete field	Administrator deletes a field. The deleted subject will also be removed from the question sets that use them.
75	Subjects	Administrator	View subjects	Administrator views the subjects created inside the system.

76	Subject detail	Administrator	View subjects detail	Administrator views the detail of a subject created inside the system.
77	Create subject	Administrator	Create new subject	Administrator creates a new subject inside the system. The learners later can use these subjects when creating new question sets.
78	Update subject	Administrator	Update subject	Administrator updates a subject.
79	Delete subject	Administrator	Delete subject	Administrator deletes a subject. The deleted subject will also be removed from the question sets that use them.

c. Screen Authorization

Screen	Learners	Guests	Administrator
Landing page		x	
Home	x	x	
Sign in	x		x
Sign up		x	
Forgot password	x		x
OTP	x	x	x
Reset password	x		x
Google authentication	x	x	x
Account	x		
Update profile	x		
Settings	x		
Confirm delete account	x		
Pomodoro	x	x	
Goals	x		
Create goal	x		
Update goal	x		

Achievements	x		
Achievement detail	x		
Explore	x	x	
Search result	x	x	
Question set detail	x	x	
Learn	x	x	
Learn options	x	x	
Practice	x		
Practice options	x		
Quiz	x	x	
Quiz options	x	x	
Matching	x	x	
Confirm rate	x		
Report question set	x		
Question comment	x	x	
Share question set	x	x	
Report comment	x		
Confirm update question set	x		
Learner profile	x	x	
Follow detail	x		
Sets	x		
Create / Update question set	x		
Create question set list	x		
Confirm delete question set	x		

Favorite question set list	x		
Groups	x		
Create group	x		
Group list	x		
Group detail	x	x	
Confirm update group	x		
Confirm delete group	x		
Confirm leave group	x		
Invitation list	x		
Send invitation	x		
Report group	x		
Dashboard			x
Administrator account			x
Administrator question sets			x
Administrator question set detail			x
Block question set			x
Unblock question set			x
Learners			x
Learner detail			x
Block learner			x
Unblock learner			x
Report			x
Report detail			x
Update report status			x

Administrator groups			x
Administrator group detail			x
Block group			x
Unblock group			x
Fields			x
Fields detail			x
Create field			x
Update field			x
Delete field			x
Subjects			x
Subject detail			x
Create subject			x
Update subject			x
Delete subject			x

In which:

- Guest: A guest is a user that has not signed up into the system. They have limited access to some features and information in the system.
- Learner: A learner is a registered user who has full access to the system. They can use all the features that the system provided.
- Administrator: An administrator is a registered user who has full access and control of the system and data. Administrator general roles include managing the platform usage and monitoring content.

d. Non-Screen Functions

#	Feature	System Function	Description
01	Appearance management	Change theme	Retrieve and store user preferred theme
02	Appearance management	Change language	Retrieve and store the user's preferred language
03	Learner management	Flag learner	Increasing the flag counter of each user by one after the Administrator attempts to flag or block the user.
04	Image management	Add image	Manage adding images when user add new image to a question set or their account profile
05	Image management	Update image	Manage updating images when user change an image from question set or their account profile
06	Image management	Delete image	Manage deleting images when user delete an image from question se or their account profile
07	Quiz result management	Calculate quiz score	Calculate the result of a quiz attempt from users.
08	Profile management	Increase learner point	Increase the learner' total point based on the type of action they achieved
09	Profile management	Synchronize data	Transfer the learner profile data between the local device and the server.
10	Recommendation	Recommend question set	Generate question set list based on popularity or learner interested fields.
11	Progress management	Track achievement progress	Keep track of learner achievement progress based on the action taken.
12	Progress management	Track event progress	Keep track of learner event progress based on the action taken.
13	Progress management	Track question set progress	Keep track of learner progress of a question set when they learn or do a quiz.

e. Entity Relationship Diagram

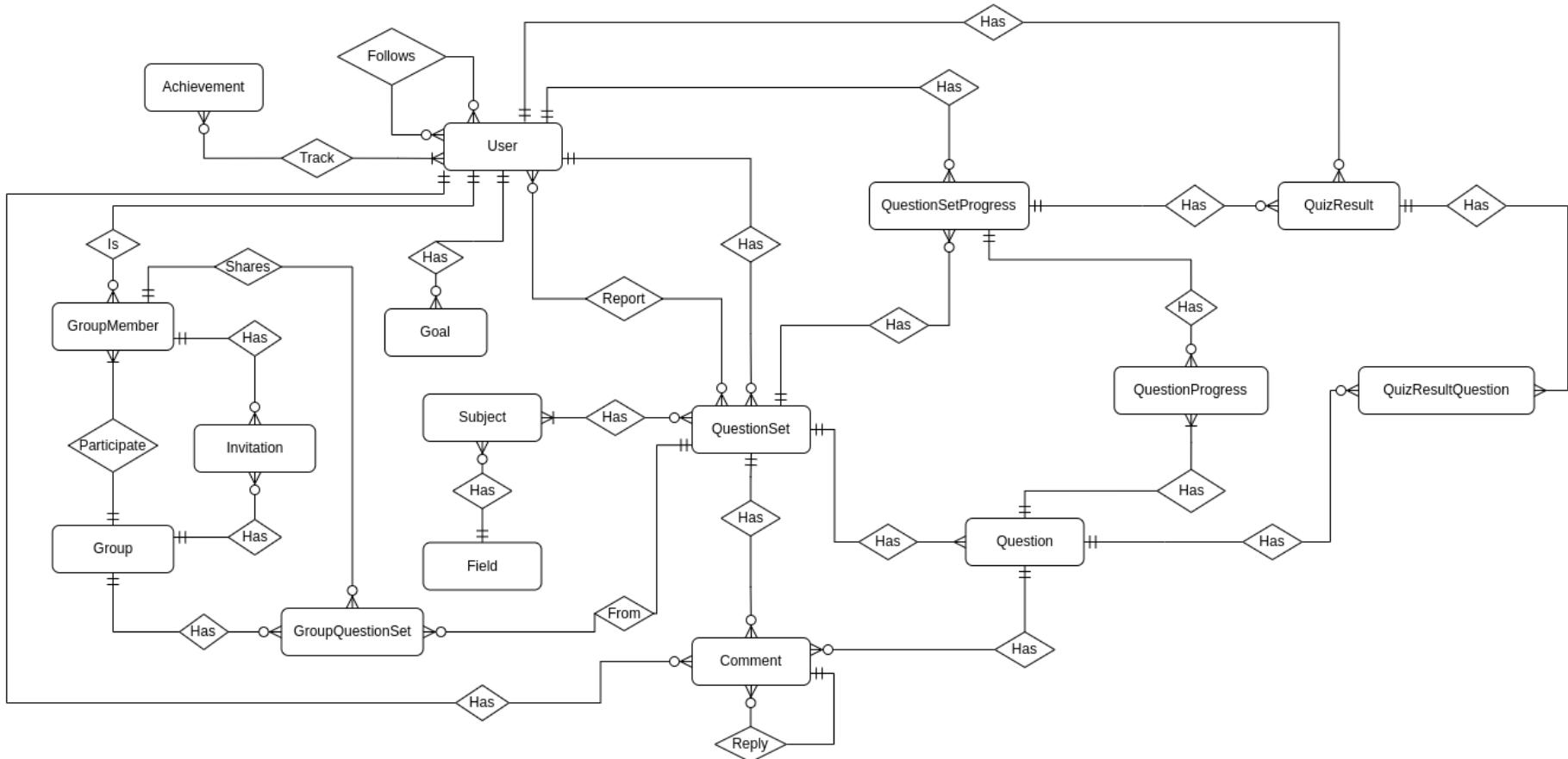


Figure 3.1.e: Entity Relationship Diagram²⁹

²⁹ [ERD.drawio.png](#)

Entities List

#	Entity	Description
01	User	Store users' information
02	Achievement	Store achievement information that learner can get
03	Goals	Store goals' information of each learner
04	QuestionSet	Store question sets' information created by learners
05	Field	Store fields information created by the Administrator
06	Subject	Store subjects information created by the Administrator
07	Comment	Store question sets comments' information, including general comments and question comments
08	Question	Store question sets questions' information
09	QuestionSetProgress	Store completion progress of each question set for each learner
10	QuestionProgress	Store question that has been learnt inside a question set, with different status for learners to filter
11	QuizResult	Store the results of each quiz attempts made by learners
12	QuizResultQuestion	Store the question that has been used for the quiz attempts, can be used for the users to revisit the attempts
13	Group	Store groups' information created by learners
14	GroupMember	Store information about each group's member
15	Invitation	Store group invitations' information sent by group members
16	GroupQuestionSet	Store question sets' information shared by group members

3.2 User authentication

3.2.1 Sign in

- *Function trigger:* User clicks on “Sign in” button
- *Function description:* Allow the user to provide their credential data, then get verified by the system to access the system.
- *Screen layout:*

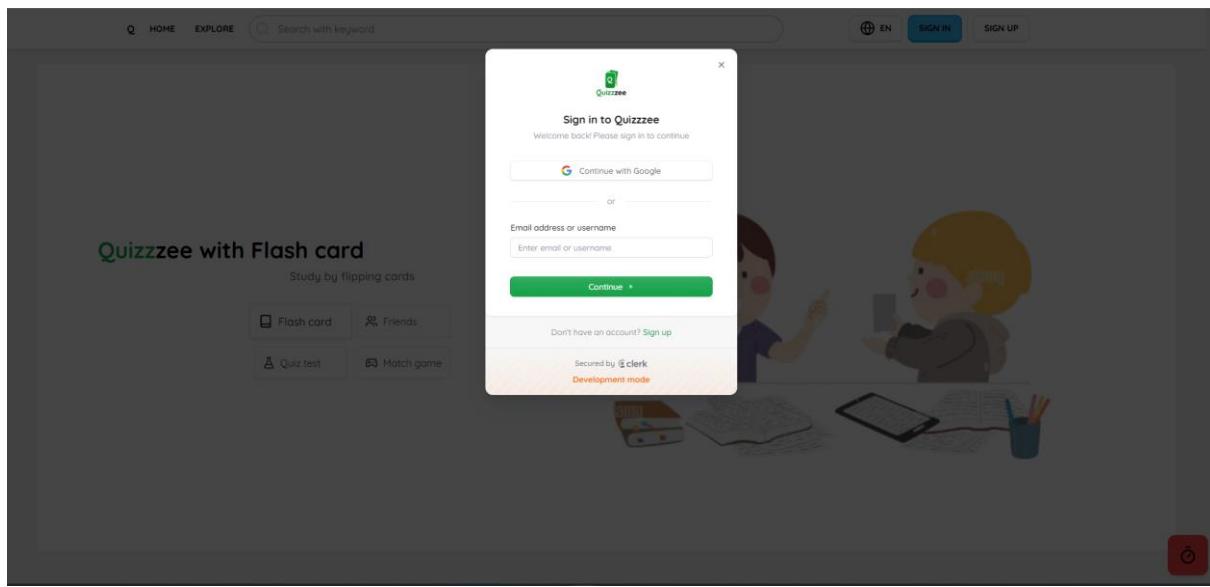


Figure 3.2.1: Sign in

- *Function detail:*
 - 1. User clicks on the “Sign in” button.
 - 2. User fills in the credential data.
 - 3. User clicks on the “Continue” button.
 - 4. The system sends the user to the homepage.

3.2.2 Sign up

- *Function trigger:* User click on “Sign up” button
- *Function description:* Allow the user to create a new account inside the system by providing credential data.
- *Screen layout:*

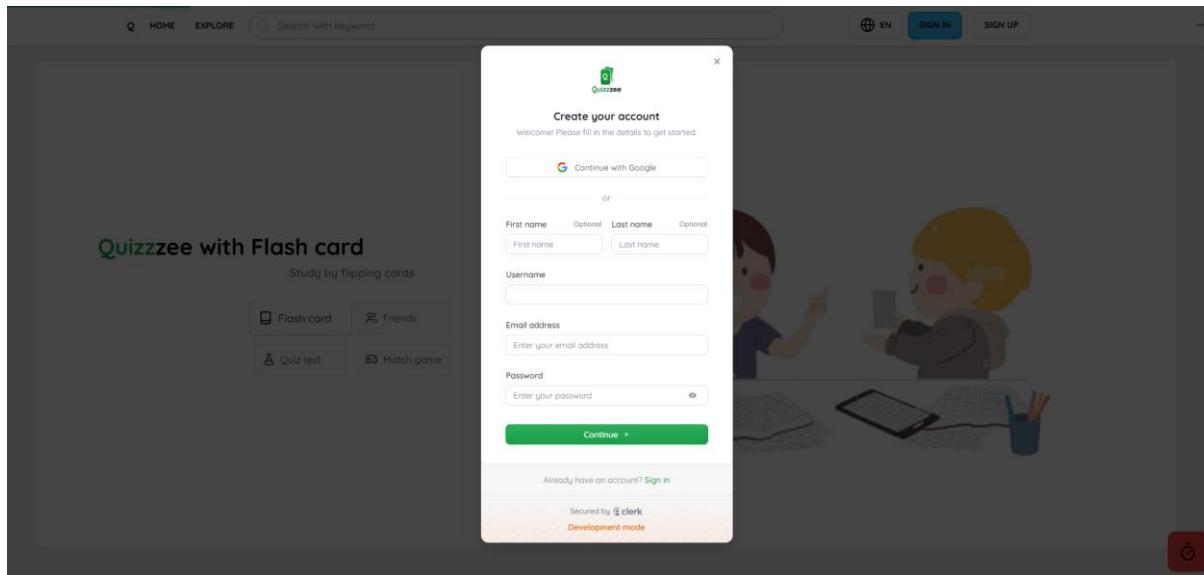


Figure 3.2.2 : Sign up

- *Function detail:*
 - 1. User clicks on the “Sign up” button.
 - 2. User fills in the credential data.
 - 3. User clicks on the “Continue” button.
 - 4. The system switches to the OTP modal.
 - 5. The system sends an OTP through user email.
 - 6. User fills in the OTP.
 - 7. User clicks on the “Continue” button.
 - 8. The system sends the user to the homepage.

3.3 Profile management

3.3.1 View account profile

- *Function trigger:* User clicks on the “Profile” button.
- *Function description:* Allow the user to view their account information
- *Screen layout:*

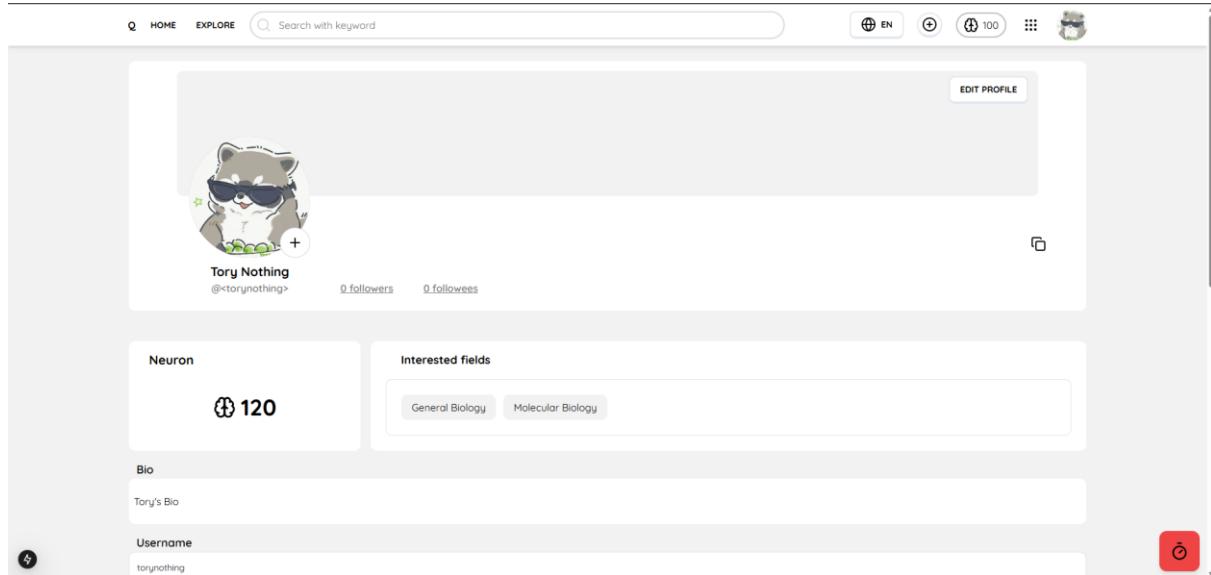


Figure 3.3.1: View account profile

- *Function detail:*
 - 1. User clicks on the “Profile” button.
 - 2. The system redirects the user to the Account page.
 - 3. The system fetches user data and displays them to the user.

3.3.2 Update account information

- *Function trigger:* User clicks on the “Edit profile” button.
- *Function description:* Allow users to update their account information such as personal data, status, field of interest.
- *Screen layout:*

The screenshot shows a user interface for editing account information. At the top left is a profile icon with '120' and the word 'Neuron'. To the right is a section titled 'Interested fields' with buttons for 'General Biology', 'Molecular Biology', and 'Add more'. Below this is a 'Bio' section containing 'Tory's Bio' and a rich text editor toolbar. The 'Username' field contains 'torynothing' with a green outline. The 'Email' field contains 'torynothing@gmail.com'. The 'Full name' field contains 'Tory Nothing'. At the bottom right is a 'SUBMIT' button.

Figure 3.3.2: Update account information

- *Function detail:*
 - 1. User clicks on the “Edit profile” button.
 - 2. User fills in their desired account information.
 - 3. User clicks on the “Update” button.
 - 4. The system verifies the data and sends it to the server.
 - 5. The system returns a popup to notify the user.
 - 6. User views the new data displayed on the Account page.

3.3.3 Update password

- *Function trigger:*
 - 1. User clicks on the “Edit profile” button from the Account page.
 - 2. User goes to the “Security” tab.
- *Function description:* Allow users to change their credential password.
- *Screen layout:*

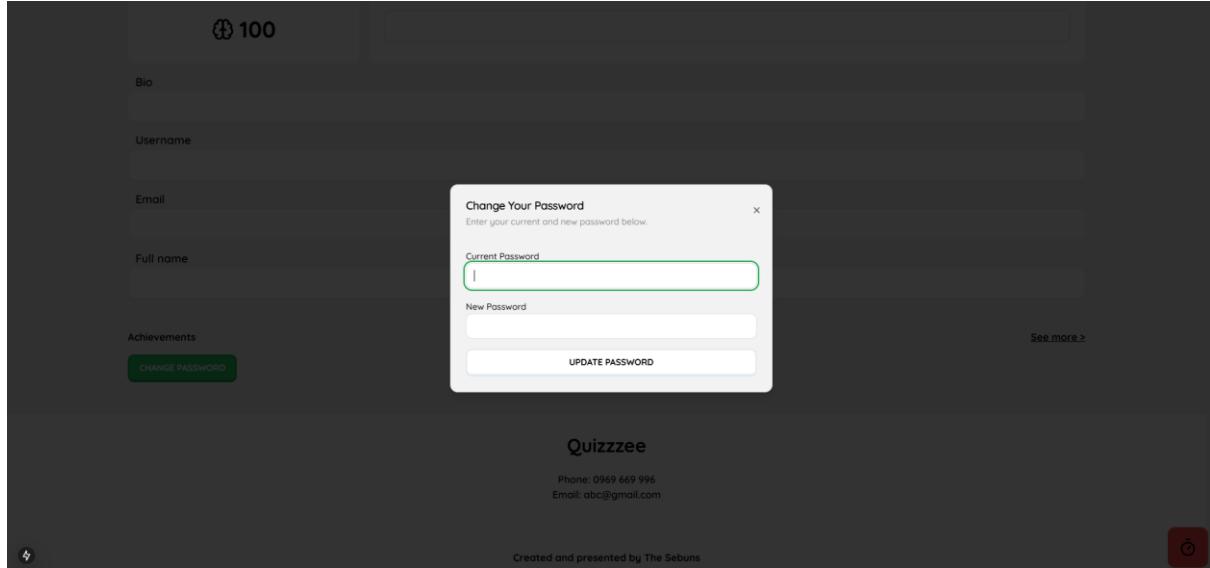


Figure 3.3.3: Update password

- *Function detail:*
 - 1. User clicks on the “Edit profile” button from the Account page.
 - 2. User goes to the “Security” tab.
 - 3. User clicks on the “Update password” button.
 - 4. User changes the password.
 - 5. User clicks on the “Save” button.
 - 6. The system updates the new password for the user.

3.3.4 Reset password

- *Function trigger:* User clicks on the “Forgot password” button on the Sign in modal
- *Function description:* Allow users to reset their password if they forgot the current one.
- *Screen layout:*

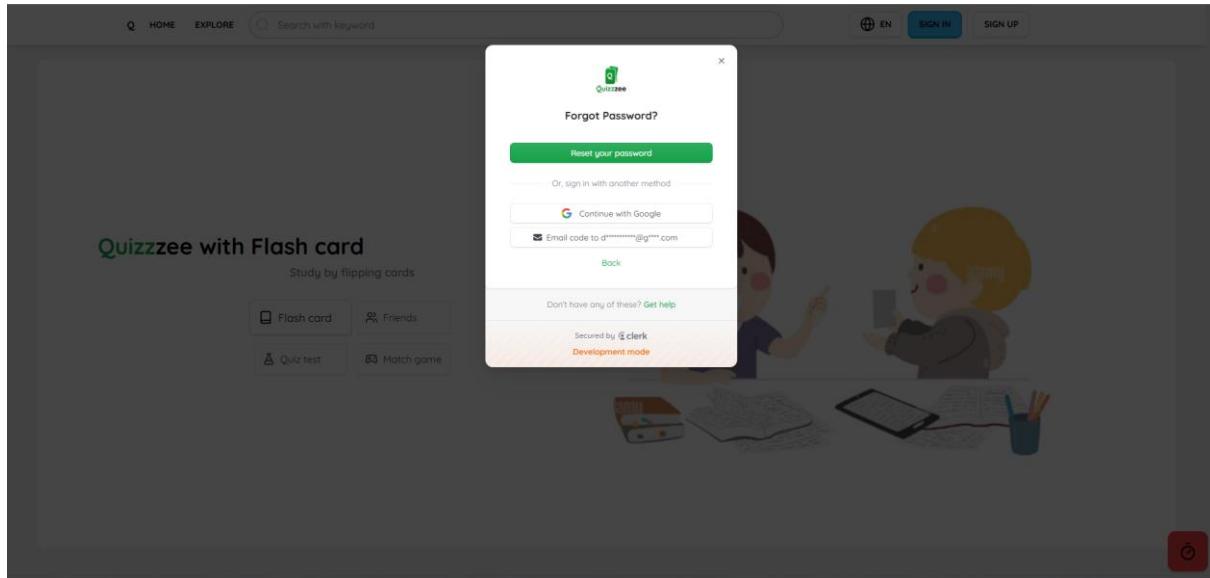


Figure 3.3.4: Reset password

- *Function detail:*
 - 1. User clicks on the “Forgot password” button on the Sign in modal.
 - 2. The system changes into Forgot password modal.
 - 3. User clicks on the “Reset password” button.
 - 4. The system changes into the OTP modal.
 - 5. The system sends an OTP code to the user email.
 - 6. User enters the OTP code.
 - 7. User change into the Reset password modal.
 - 8. The user enters the new password.
 - 9. The user clicks on the “Reset password”.
 - 10. The system verifies the password and redirects the user to the home page.

3.3.5 Delete account

- *Function trigger:* User clicks on the “Delete account” button.
- *Function description:* Allow users to permanently remove their account from the system.
- *Screen layout:*

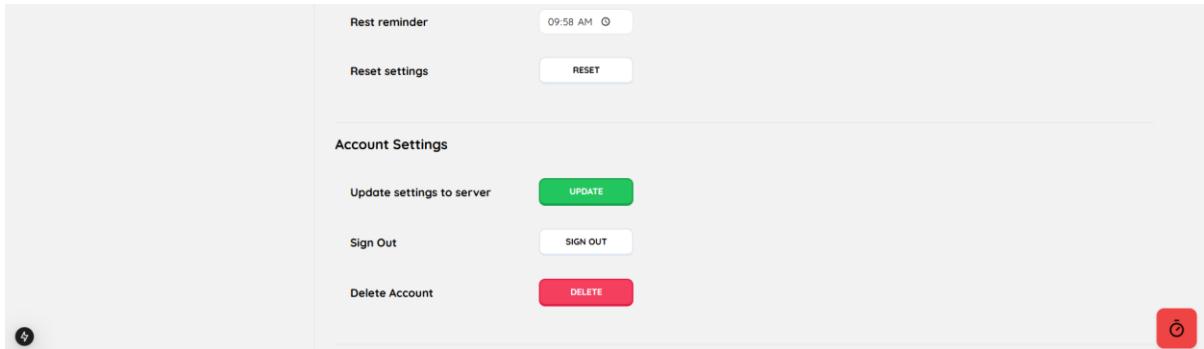


Figure 3.3.5: Delete account

- *Function detail:*
 - 1. User clicks on the “Edit profile” button from the Account page.
 - 2. User goes to the “Security” tab.
 - 3. User clicks on the “Delete account” button.
 - 4. User types “Delete account” into the field.
 - 5. User clicks the “Delete account” button.
 - 6. The system redirected the user back to the Landing page.

3.4 Question set management

3.4.1 View question sets

- *Function trigger:* User goes to the Home page
- *Function description:* Allow users to view available question sets created by other learners.
- *Screen layout:*

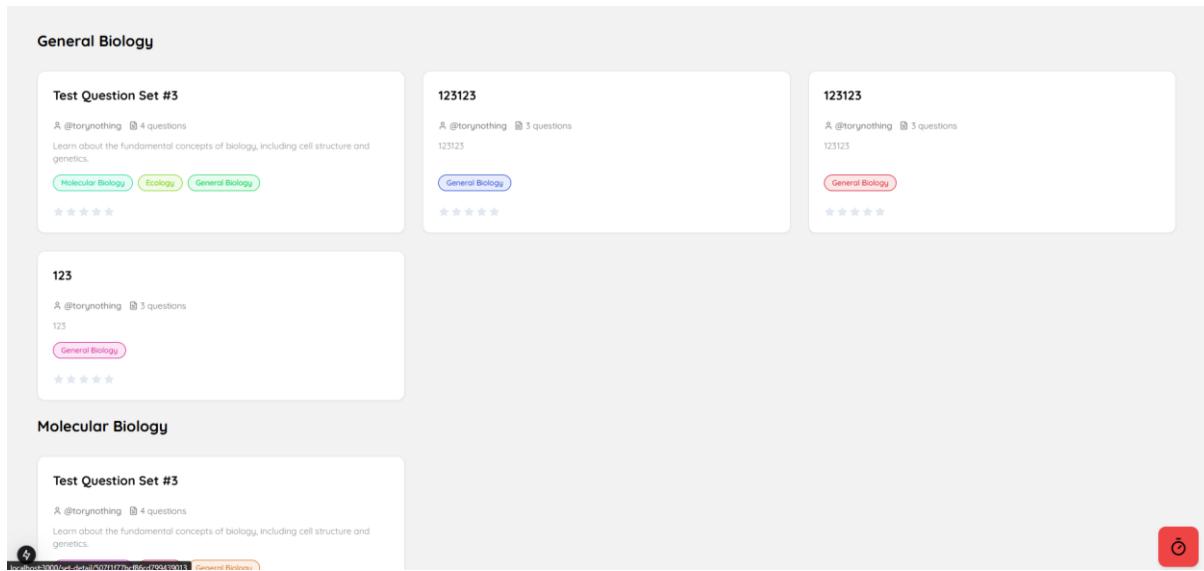


Figure 3.4.1: View question sets

- *Function detail:*
 - 1. User goes to the Home page.
 - 2. The system fetches the data from the server.
 - 3. The system displays data to the user.

3.4.2 View question set detail

- *Function trigger:* User clicks on the question set card
- *Function description:* Allow users to view the detail of a question set, including its description, questions, learners' comments,...
- *Screen layout:*

The screenshot displays a web-based application interface for viewing a question set. At the top, there's a navigation bar with links for 'HOME', 'EXPLORE', and a search bar. On the right side of the header are various icons for account management and system settings.

The main content area is titled 'Test Question Set #3'. It contains the following details:

- About this question set:** Learn about the fundamental concepts of biology, including cell structure and genetics.
- Last update:** 2025-04-16
- Rating:** ★★★★☆ (4.5 stars)
- Tags:** General Biology, Molecular Biology, Ecology

Below this, there's a user profile section for 'torynothing' with a 'FOLLOW' button. A small bio indicates they think their rating should be 4 stars, with a 5-star icon below it.

A large central area is labeled 'Learning Progress' and features a dark teal circular progress bar. To the right of the bar are four buttons: 'LEARN', 'PRACTICE', 'QUIZ', and 'PLAY'. Below the bar is a legend: Not Started (red square), Learning (green square), Still Learning (blue square), and Mastered (orange square).

Figure 3.4.2: View question set detail

- *Function detail:*
 - 1. User clicks on the question set card.
 - 2. The system redirects the user to the question set detail page.
 - 3. The system fetches the related data from the server.
 - 4. The system displays the data to the user.

3.4.3 Create question set

- *Function trigger:* User goes to the create set page.
- *Function description:* Allow users to create their own question set, and decide whether to public the set to other users.
- *Screen layout:*

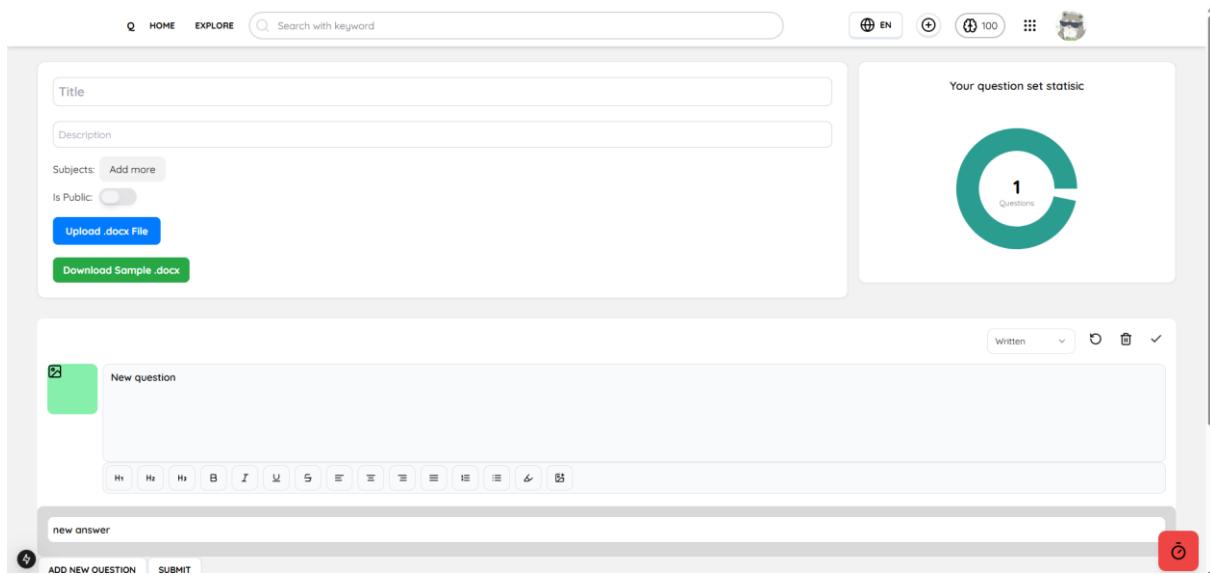


Figure 3.4.3: Create question set

- *Function detail:*
 - 1. User goes to the create set page.
 - 2. User fills in the data related to the set.
 - 3. User clicks on the “Create” button.
 - 4. The system sends the data to the server.
 - 5. The system redirects the user to the question set detail page.

3.4.4 Update question sets

- *Function trigger:* User clicks on the “Edit” button.
- *Function description:* Allow users to update their created question set.
- *Screen layout:*

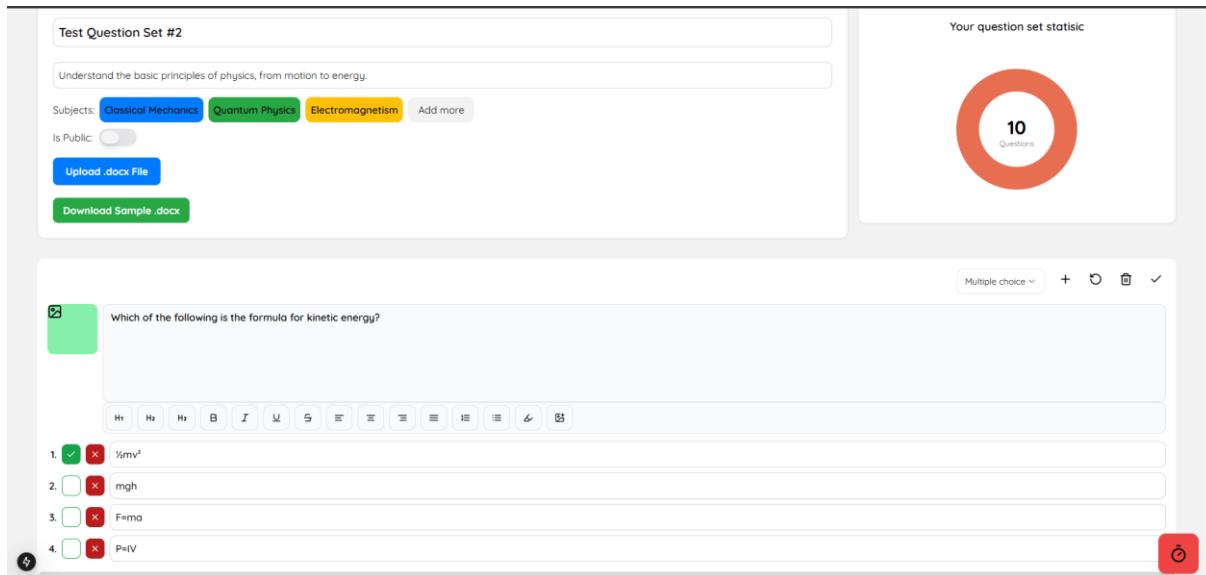


Figure 3.4.4: Update question sets

- *Function detail:*
 - 1. User clicks on the “Edit” button from the question set detail page.
 - 2. The system redirects the user to the update question set page.
 - 3. The system fetches the related data from the server.
 - 4. The system displays the data to the user.
 - 5. User changes their desired data.
 - 6. User clicks the “Update” button.
 - 7. The system sends the modified data to the server.
 - 8. The system redirects the user to the question set detail page.

3.4.5 Delete question sets

- *Function trigger:* User clicks on the delete icon.
- *Function description:* Allow users to delete their created question sets.
- *Screen layout:*

Figure 3.4.5: Delete question set

- *Function detail:*

- 1. The user clicks on the delete icon on the question set card.
- 2. The system shows the modal to confirm user action.
- 3. The user clicks on the “Delete” button.
- 4. The system sends the request to the server.
- 5. The system displayed an updated question set list to the users.

3.4.6 Share question sets

- *Function trigger:* User clicks on the “Share” button
- *Function description:* Allow users to share the question set as browser URL
- *Screen layout:*

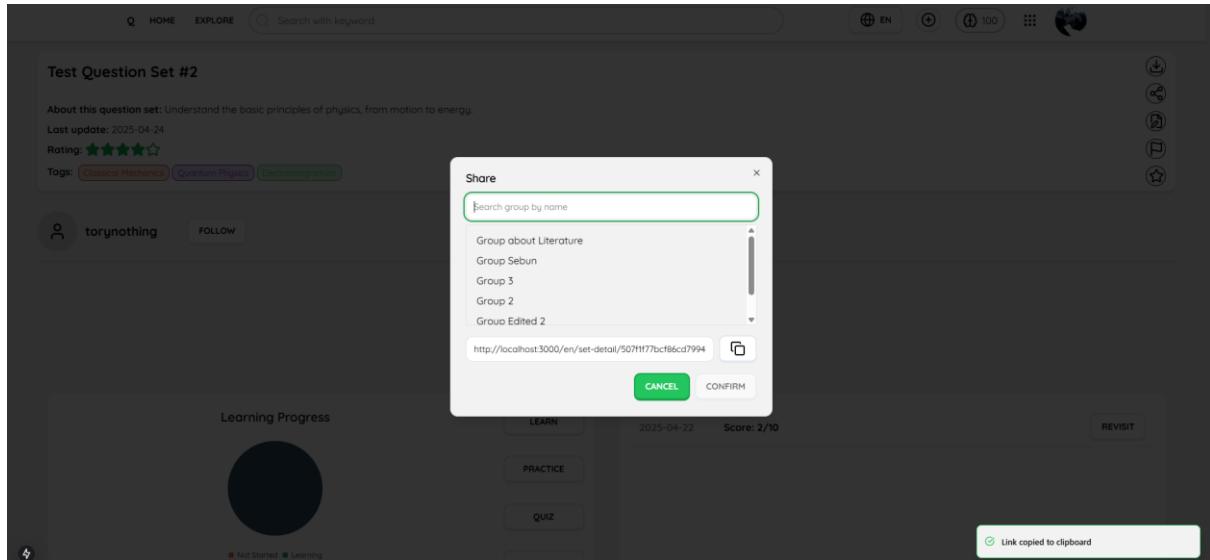


Figure 3.4.6: Share question sets

- *Function detail:*
 - 1. User clicks on the “Share” button.
 - 2. The system shows the modal displaying the question set URL.
 - 3. User clicks the copy icon.
 - 4. The question set URL is copied into the user’s system clipboard.

3.4.7 Export question sets

- *Function trigger:* User clicks on the “Export” button
- *Function description:* Allow users to export a question set to a file into their local machine
- *Screen layout:*

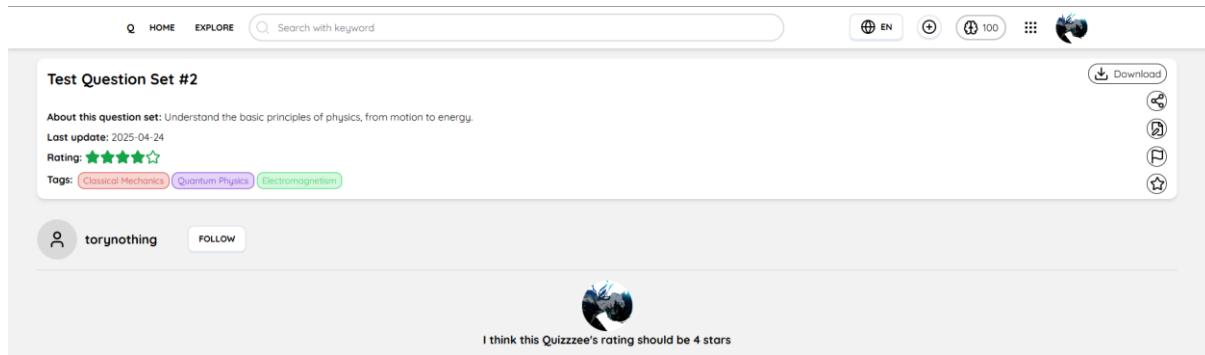


Figure 3.4.7: Export question set

- *Function detail:*
 - 1. User clicks on the “Export” button inside the question set detail page.
 - 2. The system requests the server to send a file to the client.
 - 3. The system receives file content and header from the server.
 - 4. The system initiates the download.
 - 5. The browser prompts the user to choose the download location or save automatically.
 - 6. The browser downloads the file and saves it to the user’s local machine.

3.4.8 Rate question sets

- *Function trigger:* User clicks on star icons
- *Function description:* Allow users to rate a question set by giving star
- *Screen layout:*

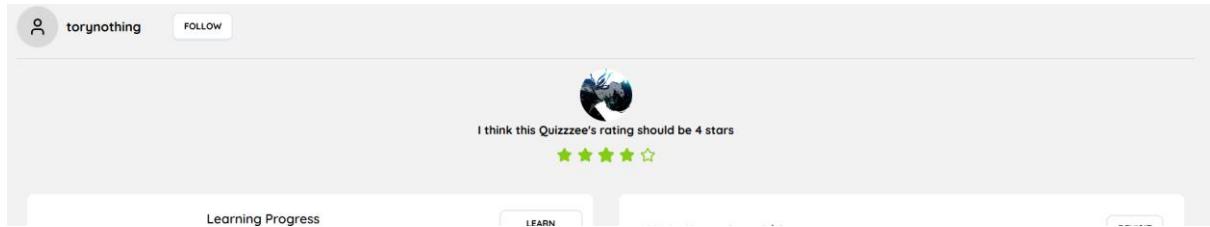


Figure 3.4.8: Rate question sets

- *Function detail:*
 - 1. User clicks on the star icons.
 - 2. The system shows the confirmation modal.
 - 3. User clicks the “Confirm” button.
 - 4. The system sends the request to the server.
 - 5. The system displays the number of stars rated by the user inside the question set detail page.

3.4.9 Toggle favourite

- *Function trigger:* User clicks on the heart icon.
- *Function description:* Allow users to add a question set to their favourite question sets.
- *Screen layout:*

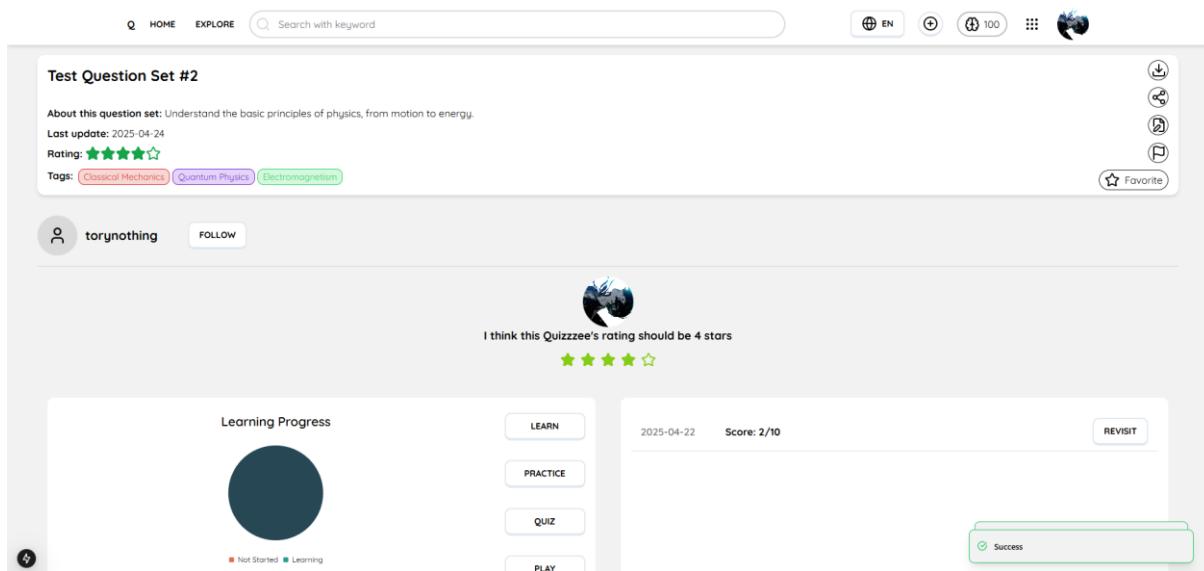


Figure 3.4.9: Toggle favourite

- *Function detail:*
 - 1. User clicks on the heart icon inside the question set detail page.
 - 2. The system sends the request to the server.
 - 3. The system updates the heart icon's look to the user.

3.4.10 Update question set accessibilities

- *Function trigger:* User click on the “Block”/“Unblock” button inside question set table row
- *Function description:* Allow the Administrator to block or unblock a question set created by learners.
- *Screen layout:*

Figure 3.4.10: Update question set accessibilities

- *Function detail:*
 - 1. User clicks on the “Block” button on the question set table row.
 - 2. The system sends the request to the server.
 - 3. The system updates the “Block” button to “Unblock” button to the user.

3.5 Group management

3.5.1 View groups

- *Function trigger:* User requests to view groups information
- *Function description:* Allow users to view groups created by learners inside the system
- *Screen layout:*

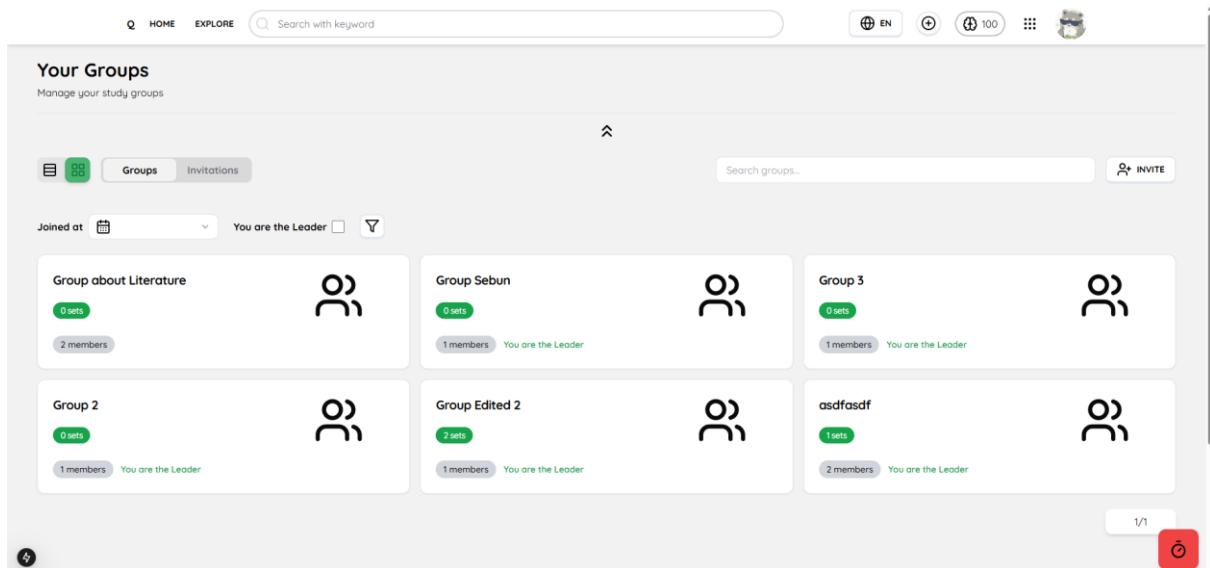


Figure 3.5.1: View groups

- *Function detail:*
 - 1. User requests to view groups information
 - 2. The system fetches data from the server.
 - 3. The system displays a list of groups to the user.

3.5.2 View group detail

- *Function trigger:* User clicks on the group card
- *Function description:* Allow users to view the detail of a groups, including participated members, shared question sets
- *Screen layout:*

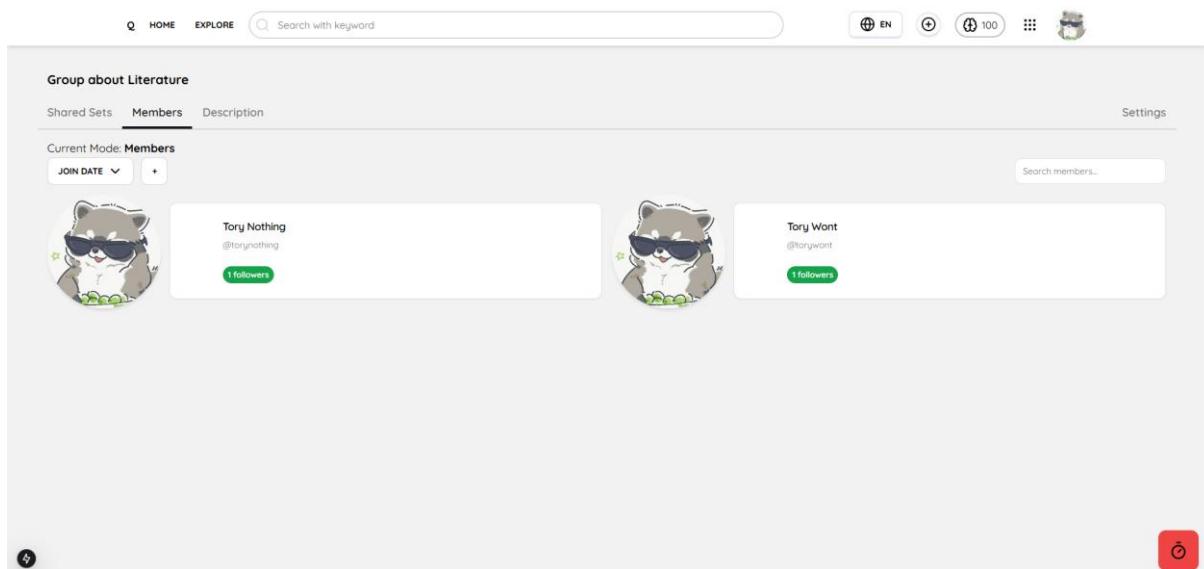


Figure 3.5.2: View group detail

- *Function detail:*
 - 1. User clicks on the group card
 - 2. The system redirects the user to the group detail page
 - 3. The system fetches the related data from the server.
 - 4. The system displays the data to the user

3.5.3 Create groups

- *Function trigger:* User goes to the create group page
- *Function description:* Allow users to create their own group to store question sets, including sharing the sets with other learners.
- *Screen layout:*

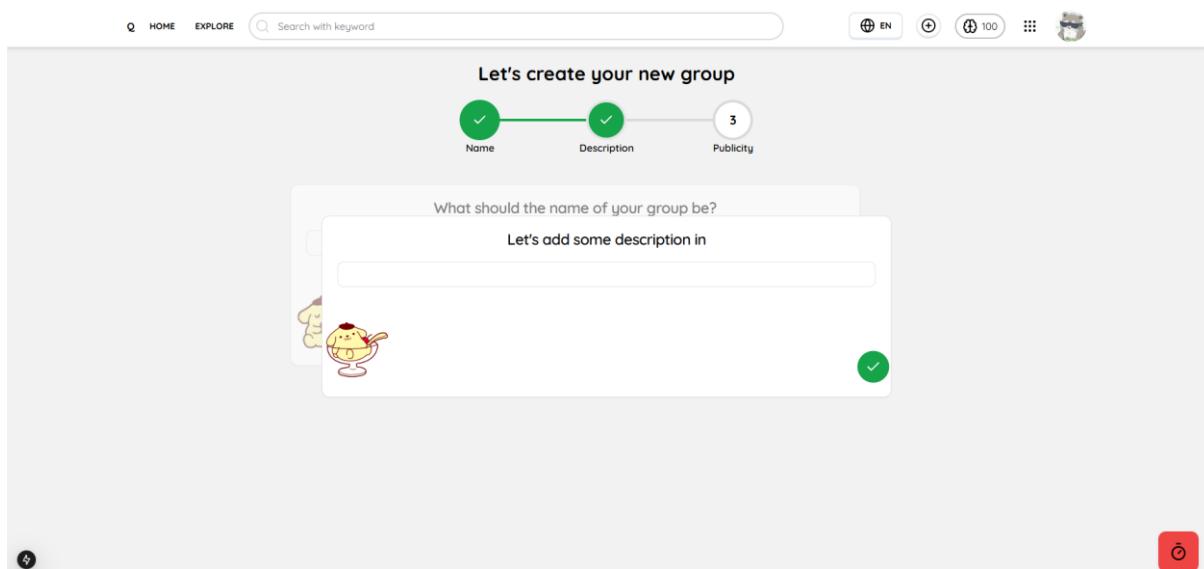


Figure 3.5.3: Create groups

- *Function detail:*
 - 1. User goes to the create group page.
 - 2. User fills in the data related to the group.
 - 3. User clicks on the “Create”button.
 - 4. The system sends the data to the server.
 - 5. The system redirects the user to the group detail page.

3.5.4 Update groups

- *Function trigger:* User clicks on the “Edit” button
- *Function description:* Allow users to update their created groups, including assigning member permissions
- *Screen layout:*

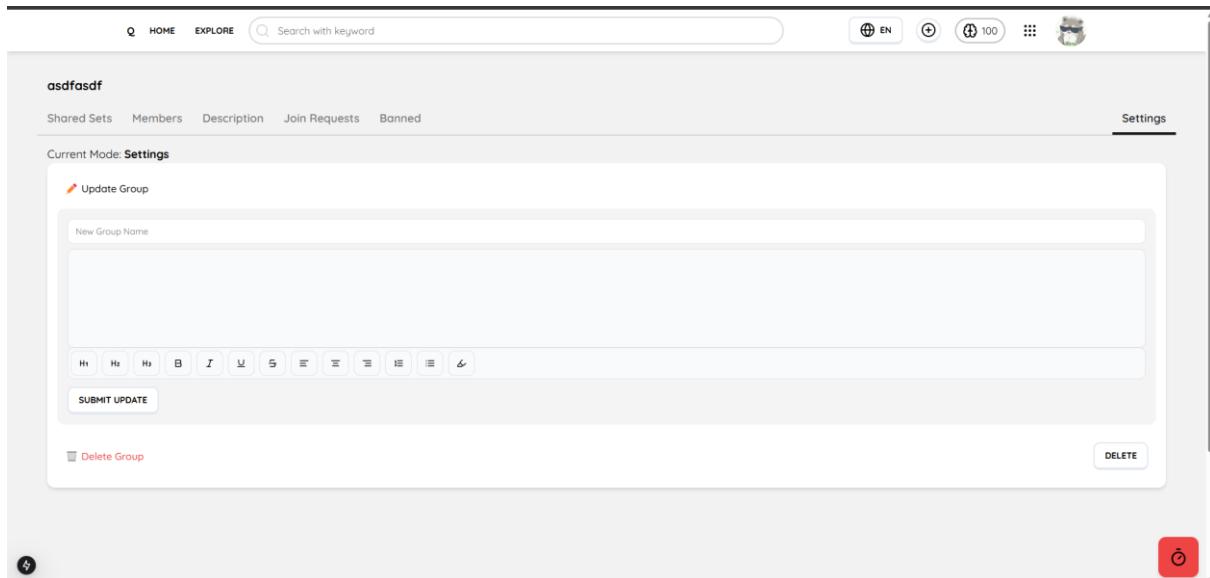


Figure 3.5.4: Update groups

- *Function detail:*

- 1. User clicks on the “Edit” button from the group detail page.
- 2. The system redirects the user to the update group page.
- 3. The system fetches the related data from the server.
- 4. The system displays the data to the user.
- 5. User changes the desired data.
- 6. User clicks on the “Update button”
- 7. The system sends the modified data to the server.
- 8. The system shows the message to notify the user about the changes.

3.5.5 Delete groups

- *Function trigger:* User clicks on the “Delete” button
- *Function description:* Allow group leaders to delete their created groups
- *Screen layout:*

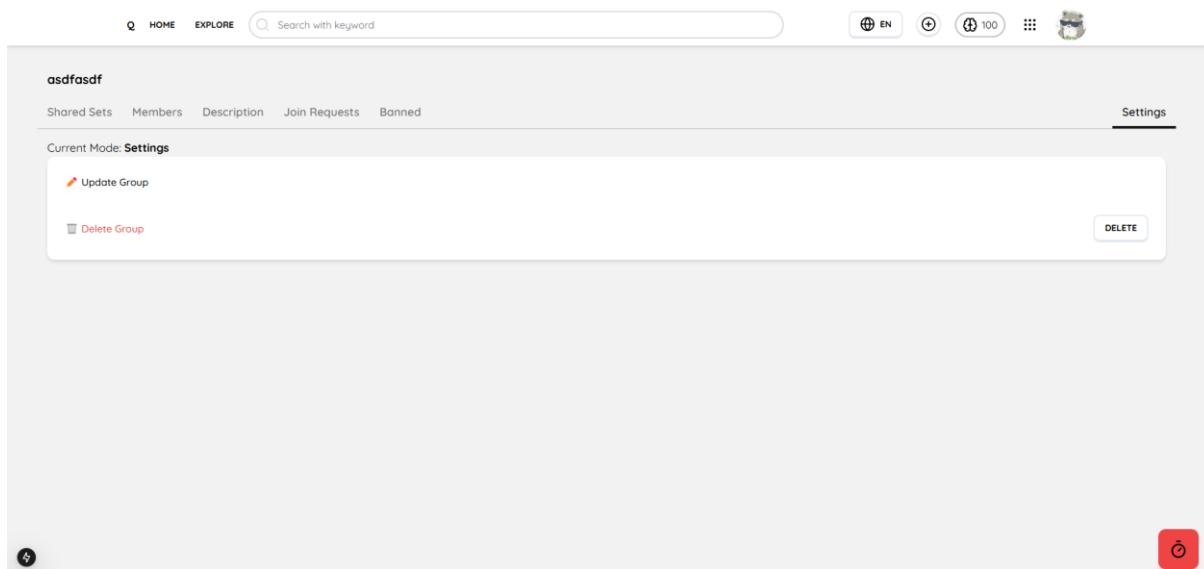


Figure 3.5.5: Delete groups

- *Function detail:*
 - 1. User clicks on the “Edit” button from the group detail page.
 - 2. The user clicks on the “Delete” button.
 - 3. The system shows the modal to confirm user action.
 - 4. The user clicks on the “Delete” button.
 - 5. The system sends the request to the server.
 - 6. The system redirects the user back to the groups page.

3.5.6 Send invitations

- *Function trigger:* User goes to the Invitation list tab
- *Function description:* Allow group members to send invitations to non-group members
- *Screen layout:*

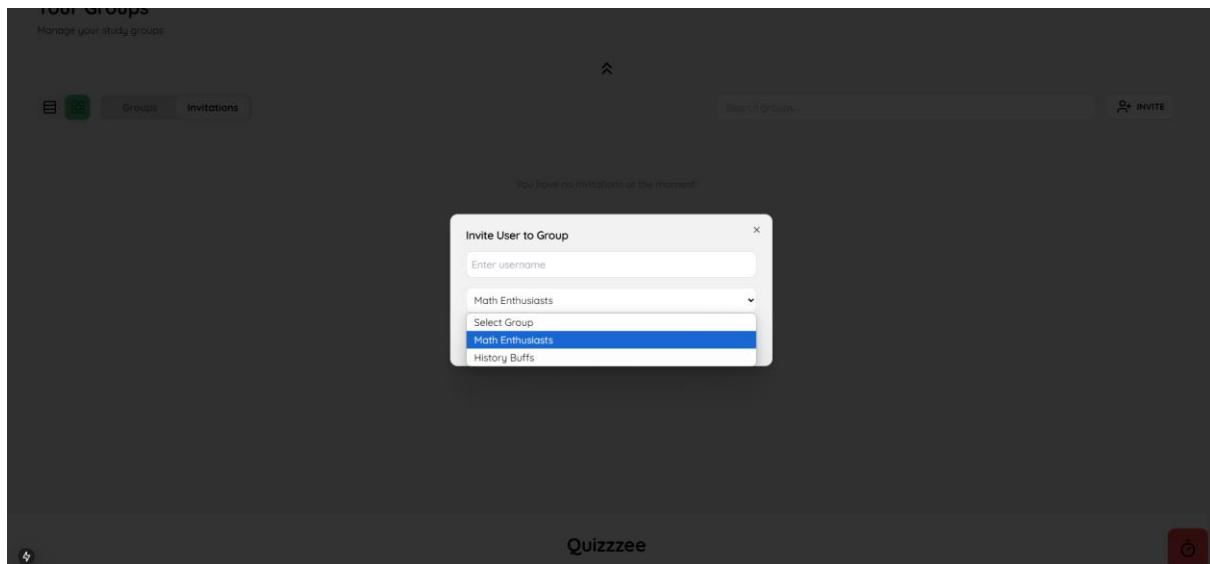


Figure 3.5.6: Send invitations

- *Function detail:*
 - 1. User goes to the Invitation list tab.
 - 2. User clicks on the “Invite” button.
 - 3. User enters a user’s username.
 - 4. User clicks on the “Invite” button.
 - 5. The system sends the data to the server.
 - 6. The system displays the message notifying the user the invitation has been sent.

3.5.7 Join groups

- *Function trigger:* User clicks on the “Join” button
- *Function description:* Allow users to join a group, including joining with invitations.
- *Screen layout:*

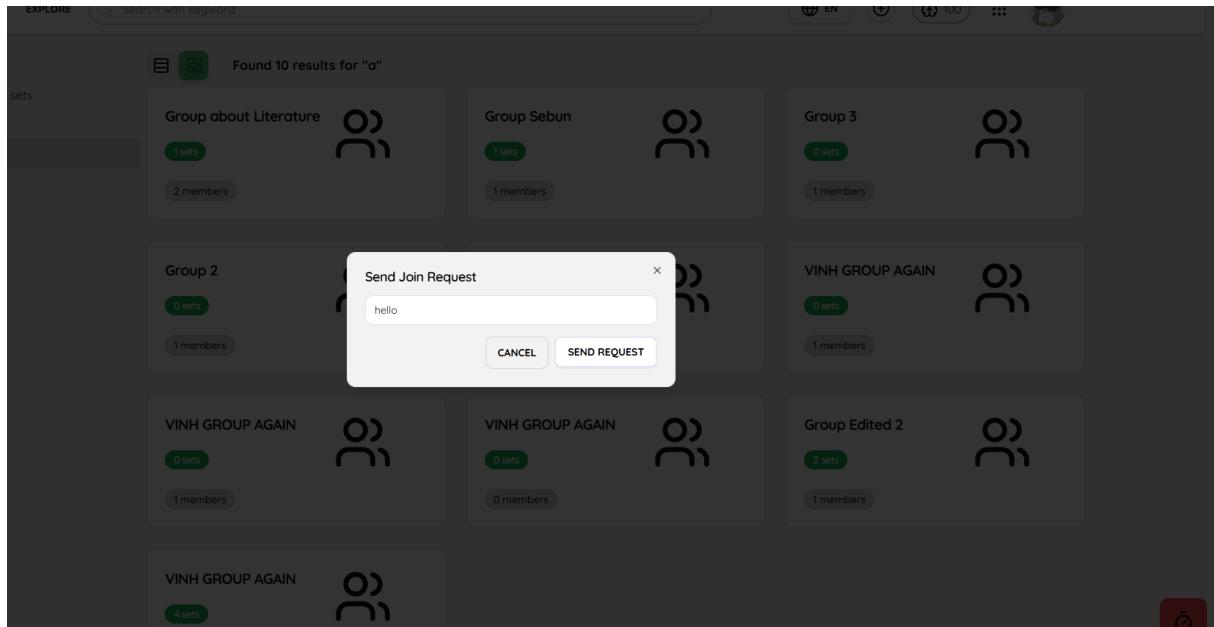


Figure 3.5.7: Join groups

- *Function detail:*
 - 1. User clicks on a group card.
 - 2. The system redirects the user to the group detail page.
 - 3. User clicks on the “Join” button.
 - 4. The system sends a request to the server.
 - 5. The system changes the “Join” button to the “Leave” button.

3.5.8 Leave groups

- *Function trigger:* User clicks on the “Leave” button
- *Function description:* Allow users to leave the current group they are participating in.
- *Screen layout:*

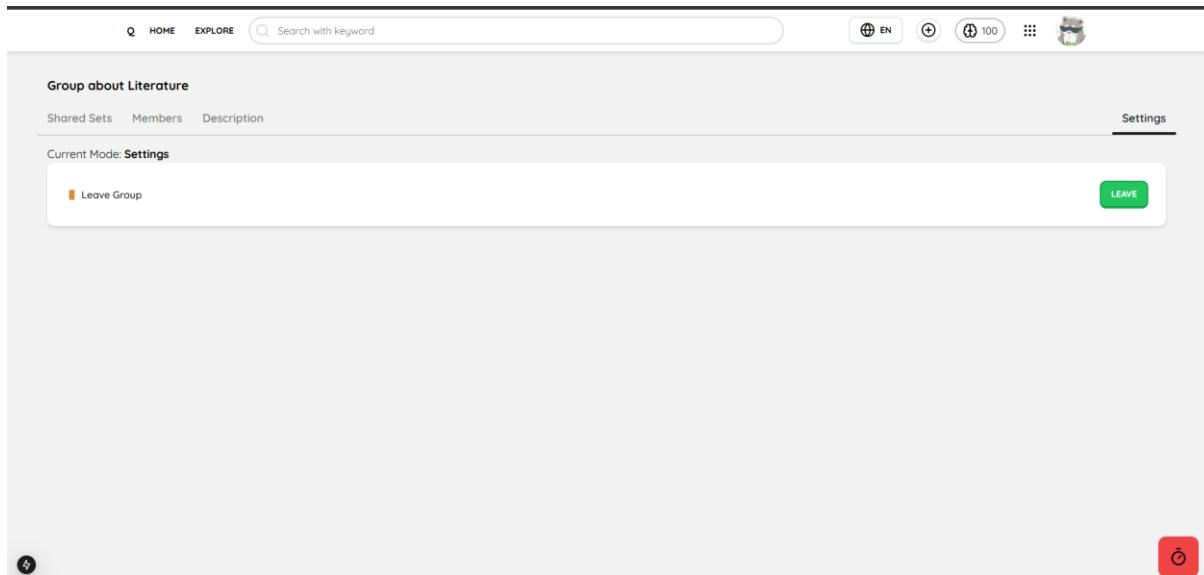


Figure 3.5.8: Leave groups

- *Function detail:*
 - 1. User goes to the group detail page.
 - 2. User clicks on the “Leave” button.
 - 3. User clicks on the “Leave” button from the modal.
 - 4. The system sends a request to the server.
 - 5. The system displays a message notifying the user that they have left the groups.
 - 6. The system changes the “Leave” button to the “Join” button.

3.5.9 Update group accessibilities

- *Function trigger:* User click on the “Block”/“Unblock” button inside group table row
- *Function description:* Allow the Administrator to block or unblock a group created by learners.
- *Screen layout:*

The screenshot shows the Quizzzee application interface. On the left, there is a sidebar with the Quizzzee logo and navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. Below these links is a user profile icon with the name "Synnothing". The main content area is titled "Group about Literature" and shows the following details:
- Group Name: Literature
- Created on: April 25, 2025
- Group Settings:

Public	Yes
Can invite	Yes
Can Add Question Sets	Yes

- Group Stats:

Members	2
Question Sets	0

- Group Leader:

	Tory Wont
--	-----------

Figure 3.5.9: Update group accessibilities

- *Function detail:*
 - 1. User clicks on the “Block” button on the group table row.
 - 2. The system sends the request to the server.
 - 3. The system updates the “Block” button to “Unblock” button to the user.

3.6 Goal management

3.6.1 View goals

- *Function trigger:* User goes to the goals page
- *Function description:* Allow users to view the goals created by themselves, this including searching and filtering for goals.
- *Screen layout:*

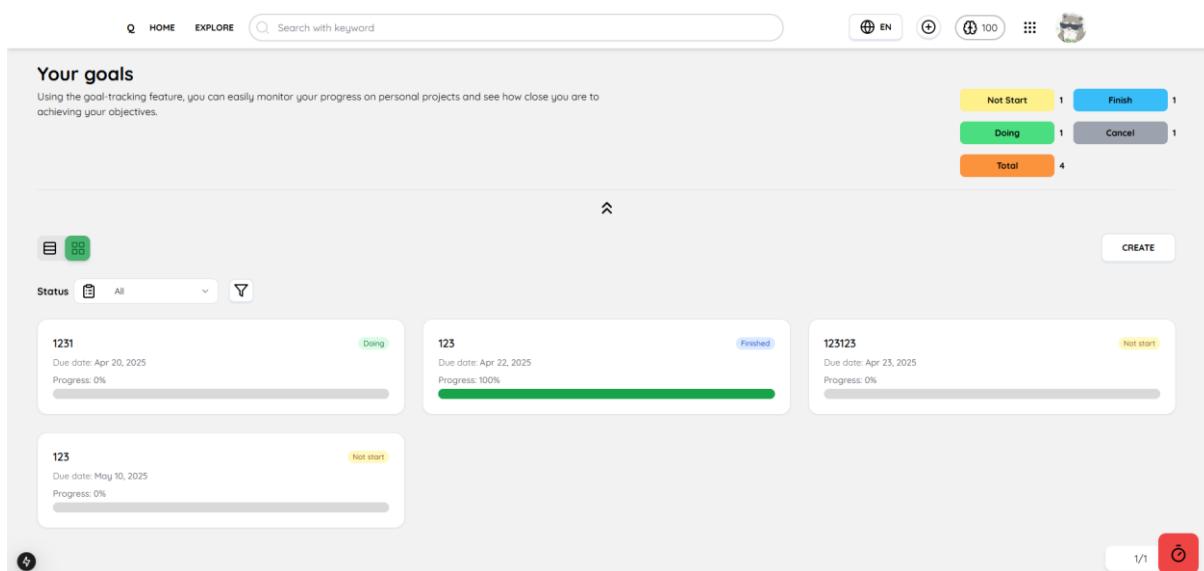


Figure 3.6.1: View goals

- *Function detail:*
 - 1. User goes to the goals page.
 - 2. The system fetches the data from the server.
 - 3. The system displays a list of created goals to the user.

3.6.2 View goal detail

- *Function trigger:* User clicks on goal card
- *Function description:* Allow users to view the details of a created goal, including current progress and deadline.
- *Screen layout:*

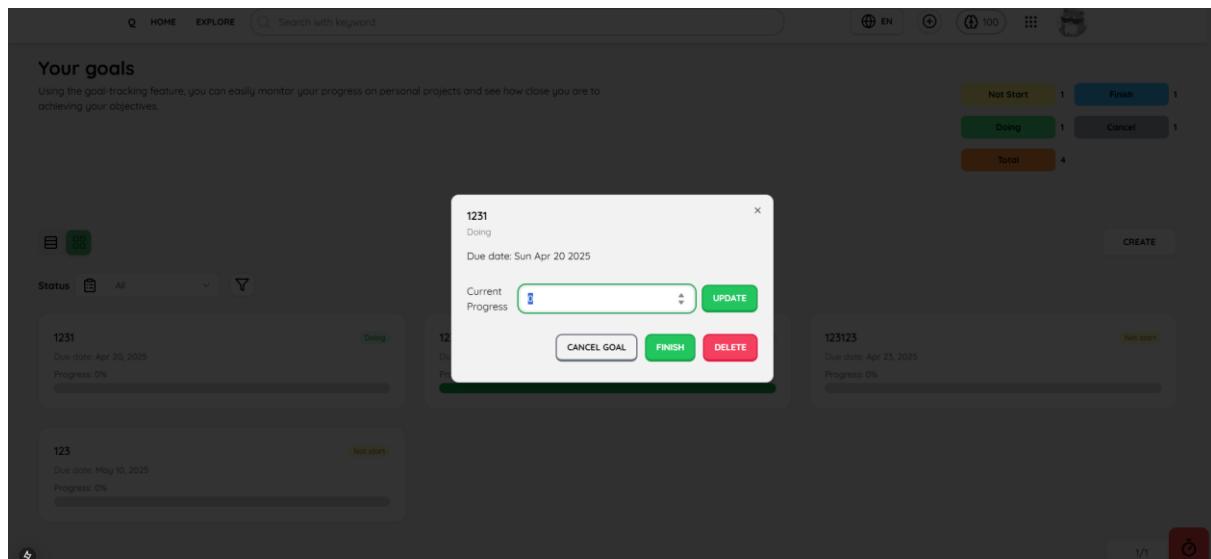


Figure 3.6.1: View goal detail

- *Function detail:*

- 1. User clicks on goal card
- 2. The system fetches the data from the server.
- 3. The system displays goal data to the user."

3.6.3 Create goals

- *Function trigger:* User clicks on the “Create” button
- *Function description:* Allow user to create their own goals
- *Screen layout:*

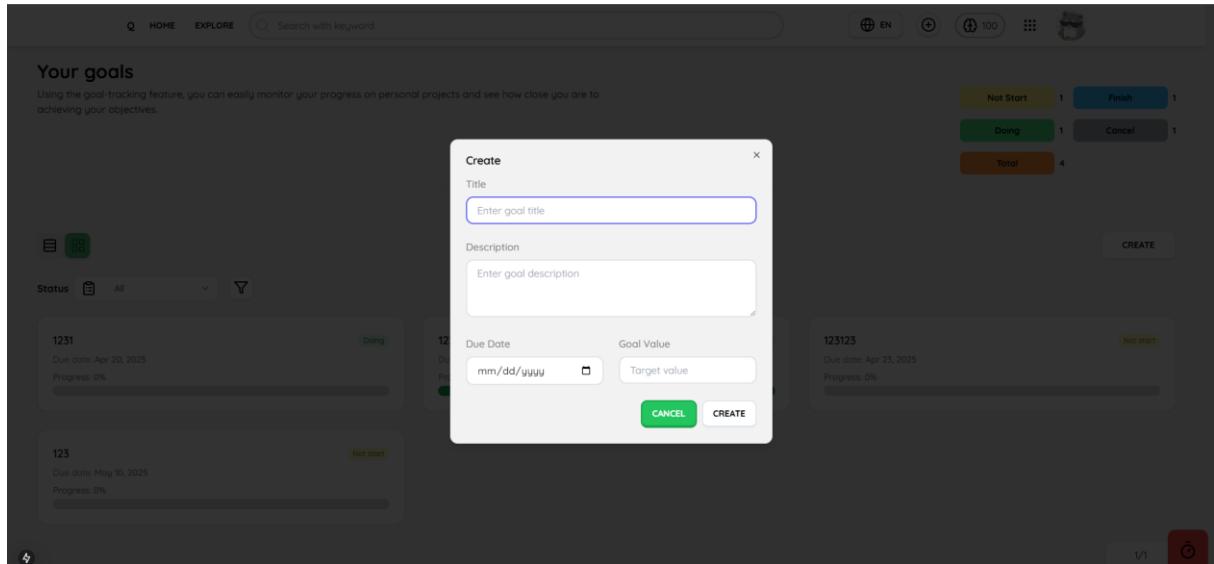


Figure 3.6.3: Create goals

- *Function detail:*
 - 1. User goes to the goals page.
 - 2. User clicks on the “Create” button.
 - 3. User fills in the necessary data.
 - 4. User clicks on the “Create” button.
 - 5. The system sends the data to the server.
 - 6. The system displays the updated list of goals to the user.

3.6.4 Update goal statuses

- *Function trigger:* User chooses a status on a goal card.
- *Function description:* Allow users to update the status of a goal, including statuses like “Finished”, “Not doing”, which told the system to stop tracking progress of goals.
- *Screen layout:*

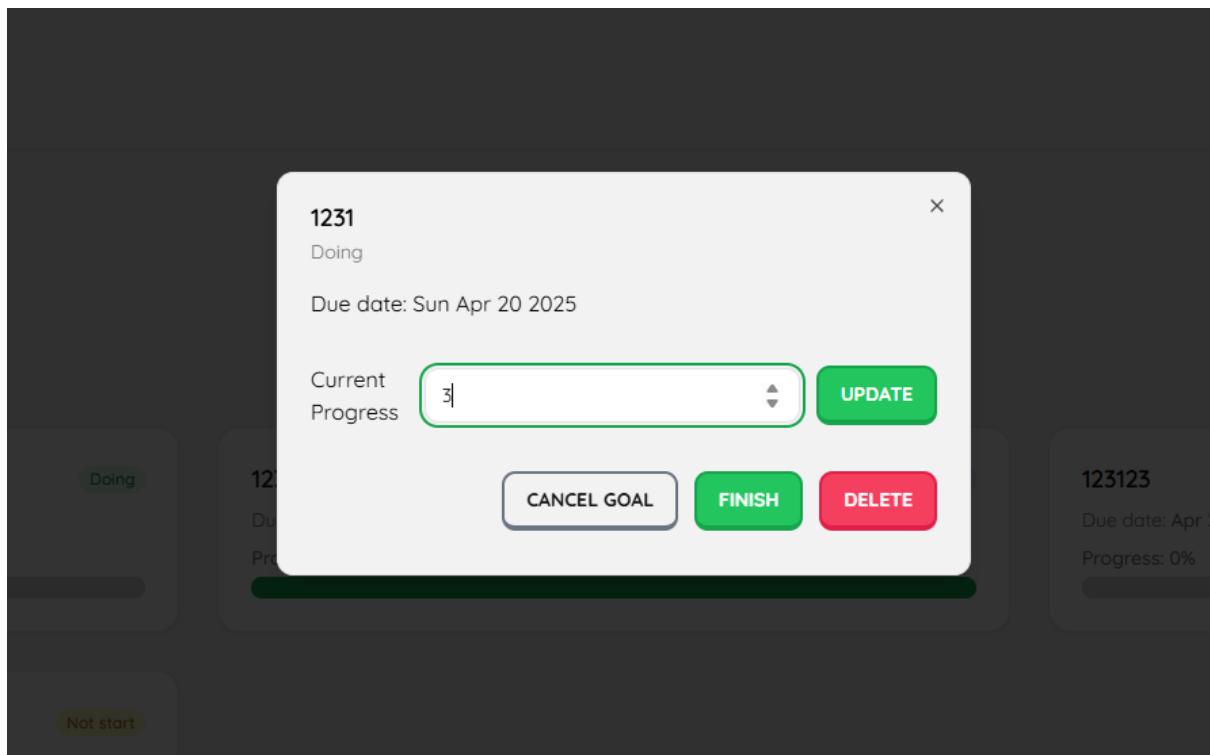


Figure 3.6.4: Update goal statuses

- *Function detail:*
 - 1. User clicks on the edit icon on the goal card.
 - 2. User clicks on a status.
 - 3. The system sends a request to the server.
 - 4. The system updates the list of goals to the user.

3.6.5 Delete goals

- *Function trigger:* User clicks on the delete button inside goal detail modal
- *Function description:* Allow users to delete their created goals.
- *Screen layout:*

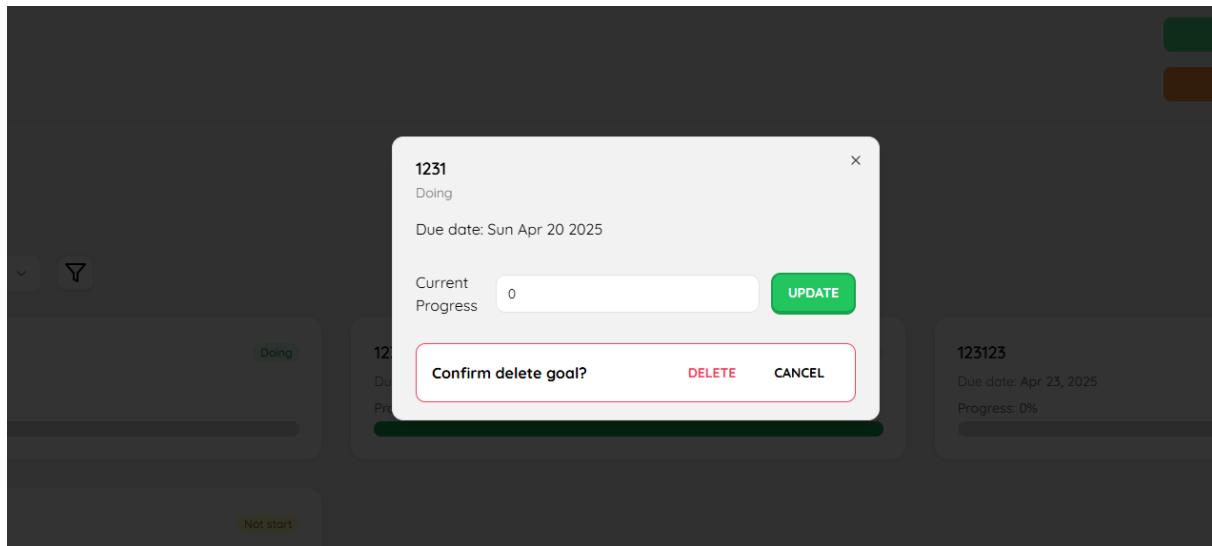


Figure 3.6.5: Delete goals

- *Function detail:*
 - 1. User clicks on the delete icon.
 - 2. User clicks on the “Delete” button inside the confirm modal.
 - 3. The system sends a request to the server.
 - 4. The system updates the list of goals to the user.

3.7 Setting management

3.7.1 View settings

- *Function trigger:* User clicks on the “Settings” button
- *Function description:* Allow users to view the available system settings.
- *Screen layout:*

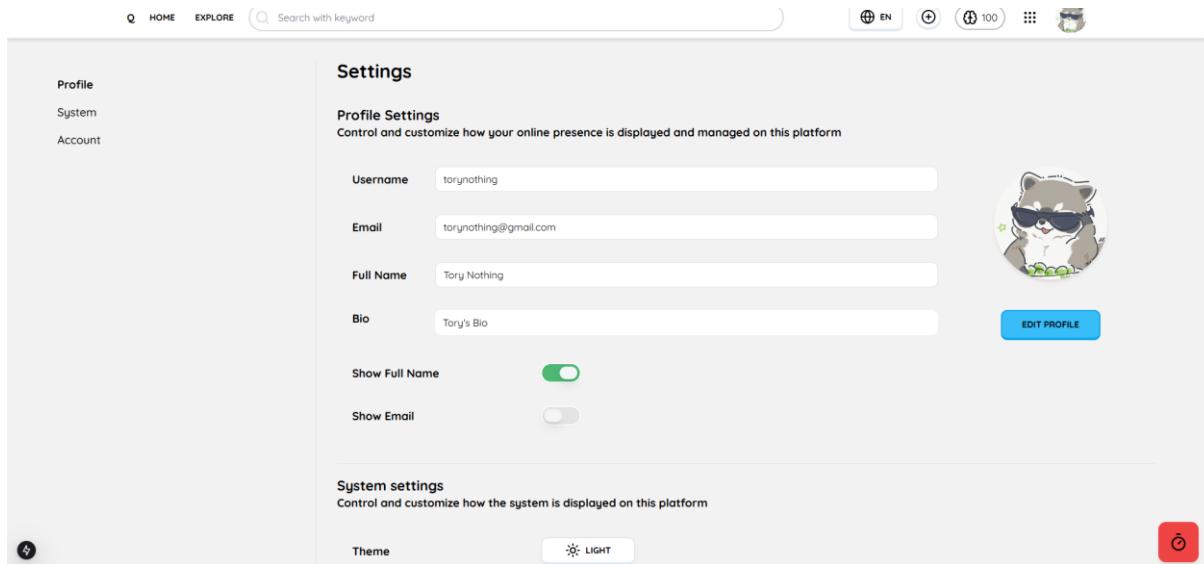


Figure 3.7.1: View settings

- *Function detail:*
 - 1. User clicks on the “Settings” button.
 - 2. The system redirects the user to the settings page.
 - 3. The system fetches user customized settings from the server.
 - 4. The system displays all the settings options to the user.

3.7.2 Update system appearance

- *Function trigger:* User change a setting option
- *Function description:*
- *Screen layout:*

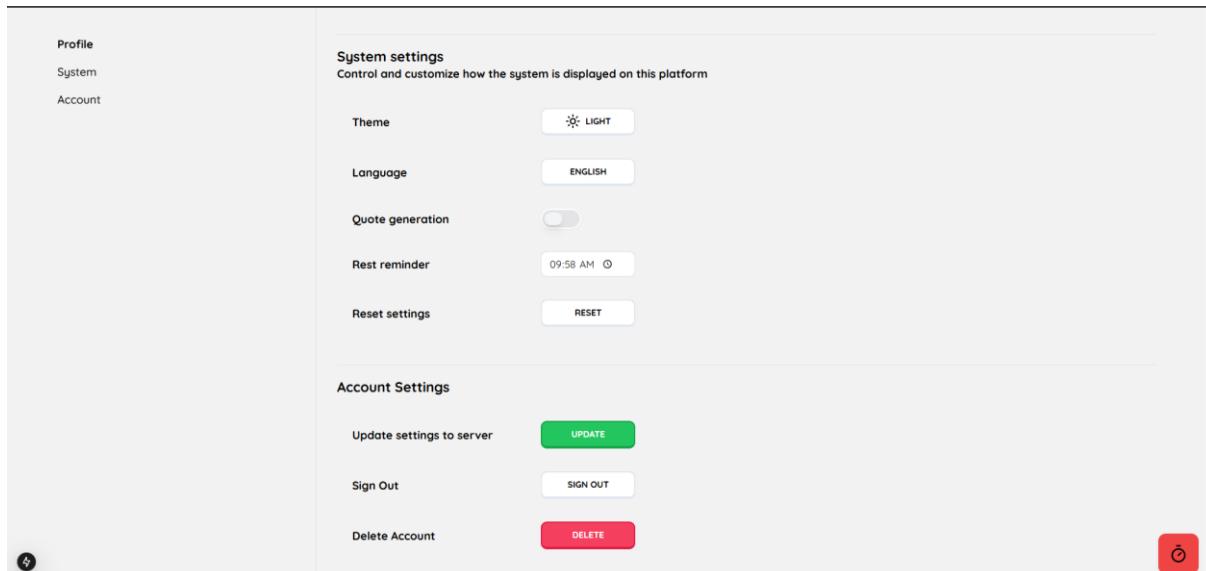


Figure 3.7.2: Update system appearance

- *Function detail:*
 - 1. User clicks on the language dropdown.
 - 2. User chooses one of the options “English” or “Tiếng Việt”.
 - 3. The system updates the appearance based on the option changes.
 - 4. The user clicks on the “Save changes” button.
 - 5. The system sends a request to the server.
 - 6. The system displays a message notifying the user.

3.7.3 Set rest reminder

- *Function trigger:* The user sets a specific time for the reminder.
- *Function description:* The system allows users to set a specific time for rest reminders. When the set time is reached, the user will automatically receive a notification prompting them to take a break.
- *Screen layout:*

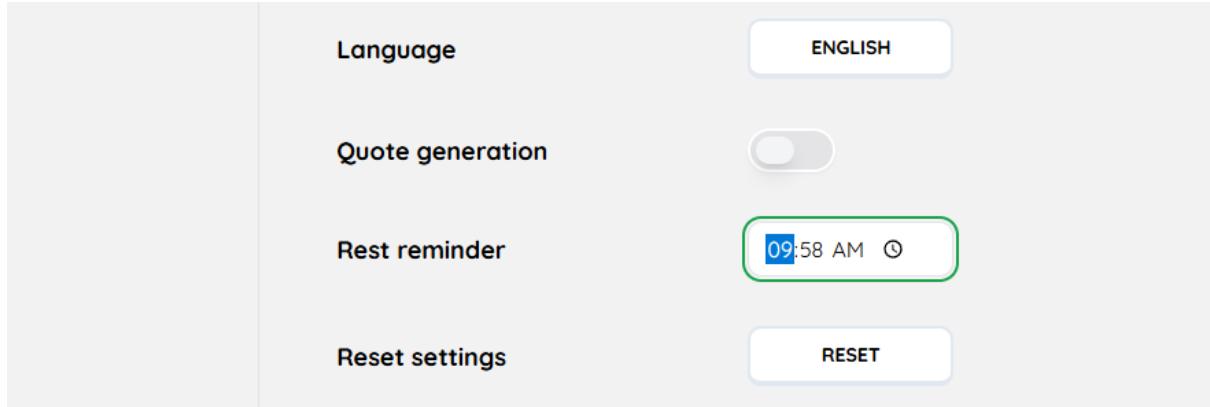


Figure 3.7.3: Set rest reminder

- *Function detail:*
 - 1. User goes to the settings page.
 - 2. User sets a specific timer.
 - 3. User clicks on the “Save changes” button.
 - 4. The system sends a request to the server.
 - 5. The system displays a message notifying the user.

3.7.4 Reset settings

- *Function trigger:* User clicks on the “Reset” button
- *Function description:* Allow users to reset their custom settings back to default options
- *Screen layout:*

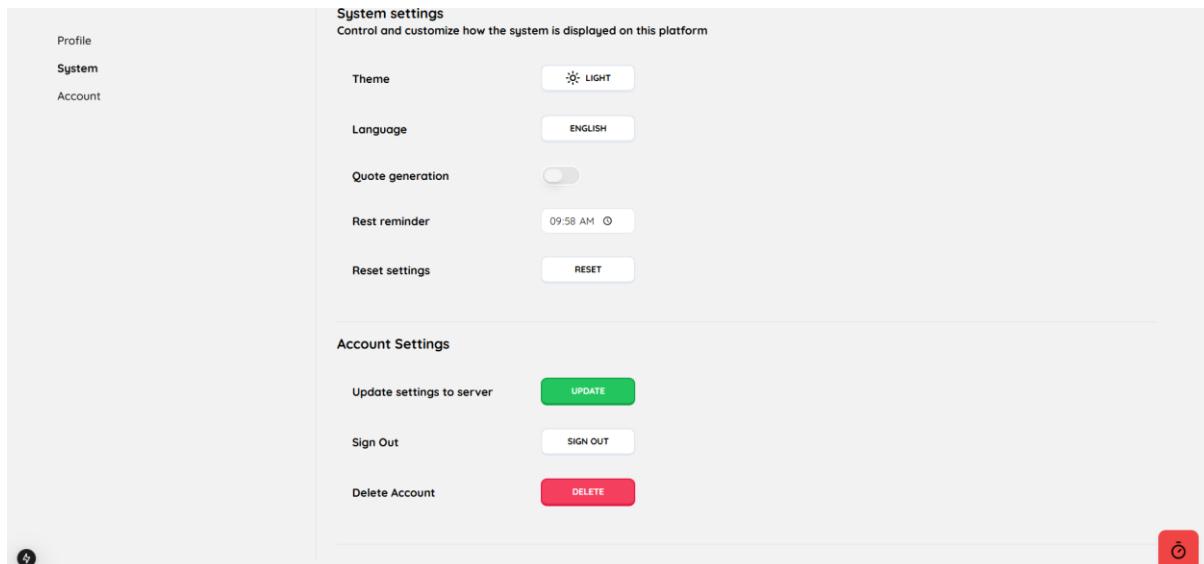


Figure 3.7.4: Reset settings

- *Function detail:*
 - 1. User clicks on the “Reset” button.
 - 2. The system displays a modal asking the user to confirm resetting.
 - 3. User clicks on the “Reset” button.
 - 4. The system sends a request to the server.
 - 5. The system displays a message notifying the user.

3.8 Learner management

3.8.1 View learners

- *Function trigger:* User goes to pages with learners displayed.
- *Function description:* Allow users to view learners who have valid accounts inside the system.
- *Screen layout:*

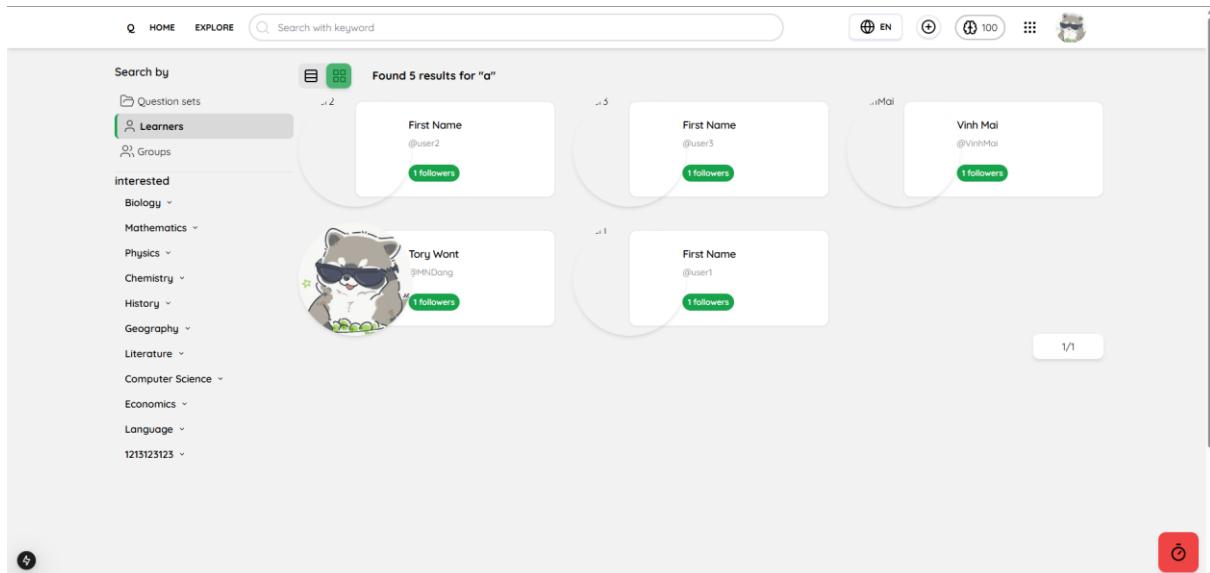


Figure 3.8.1: View learners

- *Function detail:*
 - 1. User goes to the home page.
 - 2. The system fetches data from the server.
 - 3. The system displays a list of users to the user.

3.8.2 View learner profile

- *Function trigger:* User goes to the learner profile page
- *Function description:* Allow users to view other learner profile data. Including following status, personal information, created question sets or participating public groups.
- *Screen layout:*

The screenshot shows a learner profile page for a user named 'Tory Wont' (@MN_Dang). The profile includes a cartoon raccoon icon with sunglasses, a bio section with 'Test bio #2', follower counts (0 followers, 0 followees), and a 'FOLLOW' button. Below this are sections for 'Interested Fields' (Database Systems, Calculus, Statistics) and 'Achievements' (TODO). The main area displays a grid of nine 'Test Question Set' cards, each with a title, author, question count, description, and tags. For example, 'Test Question Set #3' is by @torynothing with 4 questions, covering Molecular Biology, Ecology, and General Biology. Other sets include Test Question Set #5 (English grammar), Test Question Set #6 (World History, Ancient Civilizations), Test Question Set #2 (Physics), Test Question Set #8 (C++ programming), Test Question Set #4 (Computer Science), Test Question Set #7 (Electromagnetism), Test Question Set #9 (Organic chemistry), and a set by user 123123.

Figure 3.8.2: View learner profile

- *Function detail:*
 - 1. User goes to the learner profile page.
 - 2. The system fetches the data from the server.
 - 3. The system displays data related to the learner to the user.

3.8.3 Update learner accessibilities

- *Function trigger:* User click on the “Block”/“Unblock” button inside learner table row
- *Function description:* Allow the Administrator to block or unblock a learner account.
- *Screen layout:*

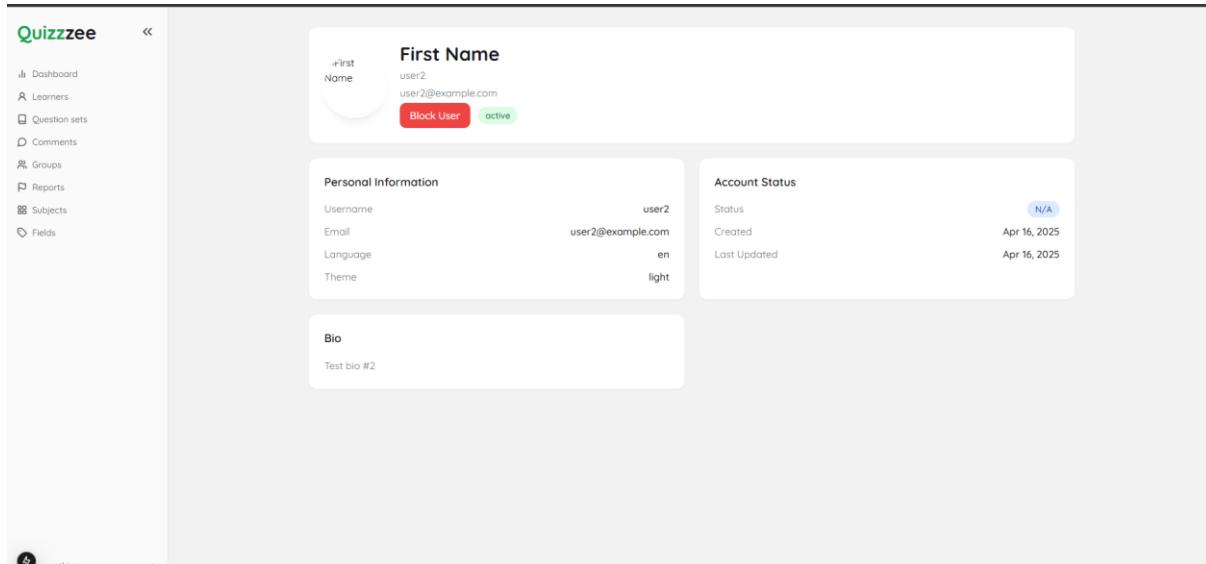


Figure 3.8.3: Update learner accessibilities

- *Function detail:*
 - 1. User clicks on the “Block” button on the learner table row.
 - 2. The system sends the request to the server.
 - 3. The system updates the “Block” button to “Unblock” button to the user.

3.9 Report management

3.9.1 Create reports

- *Function trigger:* User clicks on the “Report” button.
- *Function description:* Allow users to report the inappropriate question set, comments or groups
- *Screen layout:*

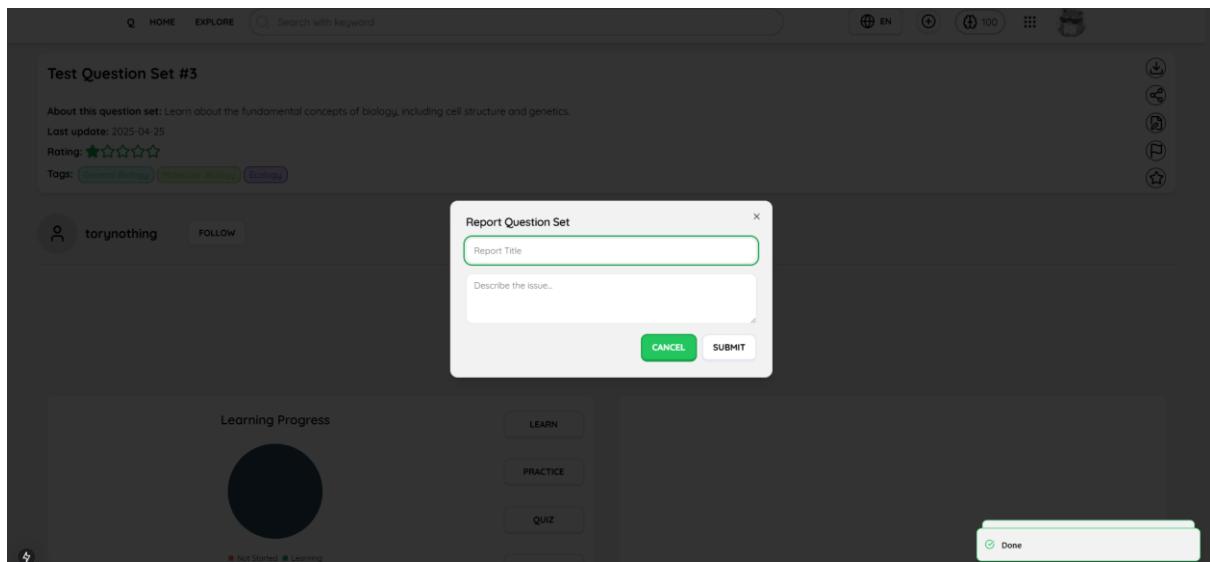


Figure 3.9.1: Report question sets

- *Function detail:*
 - 1. User clicks on the “Report” button
 - 2. The system displays a report modal.
 - 3. User fills in the description of the report.
 - 4. User clicks on the “Report” button.
 - 5. The system sends the data to the server.
 - 6. The system displays a message to notify the user."

3.9.2 View reports

- *Function trigger:* User goes to the reports page.
- *Function description:* Allow the Administrator to view user reports sent by learners.
- *Screen layout:*

The screenshot shows the Quizzzee application interface. On the left is a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports (which is selected), Subjects, and Fields. The main area is titled "Reports" and shows a table of 13 reports found. The table has columns: Reported By, TYPE, Title, Description, Resolved, and Created At. A filter bar at the top allows filtering by username. Navigation buttons for PREVIOUS and NEXT are at the bottom right of the table area.

Reported By	TYPE	Title	Description	Resolved	Created At
torynothing	comment	osdf	asdfasdf	No	4/18/2025, 4:24:19 AM
torynothing	comment	osdf	asdfasdf	No	4/18/2025, 4:25:34 AM
torynothing	comment	dasdasdasdasd	asdasdasdas	No	4/18/2025, 4:31:47 AM
torynothing	user	asd	osdasd	No	4/18/2025, 4:47:32 AM
torynothing	questionSet	asd	osdasdasd	Yes	4/18/2025, 5:29:07 AM
torynothing	questionSet	asdas	dasdasdasd	Yes	4/18/2025, 6:16:00 AM
torynothing	questionSet	TESTINGGGGGGGGG	osdasdasdasd	Yes	4/18/2025, 9:47:28 AM
torynothing	comment	osdf	asdfasdf	No	4/21/2025, 9:40:43 PM
torynothing	comment	osdf	asdfasdf	No	4/21/2025, 9:48:55 PM
torynothing	comment	osdf	asdfasdf	No	4/21/2025, 10:11:38 PM

Figure 3.9.2: View reports

- *Function detail:*
 - 1. User goes to the reports page.
 - 2. The system fetches the data from the server.
 - 3. The system displays a table of created reports to the user.

3.9.3 View report detail

- *Function trigger:* User goes to the report detail page
- *Function description:* Allow the Administrator to view report detail sent by learners.
- *Screen layout:*

The screenshot shows the Quizzzee application interface. On the left, there's a sidebar with a navigation menu. The main content area is titled 'Report Detail' and shows a report with the ID 'asdf'. It includes fields for 'Reported by' (torynothing), 'Status' (Pending), and a 'Mark as Resolved' button. Below this is a 'Comment Information' section for a comment from user 'VinhMai' (User ID: user_2rOZJvHNv1ZHJHRSpBhc0D4lx). The comment content is 'This question needs more clarity in the wording.' and it has 5 upvotes. The comment metadata shows it was created and updated on April 16, 2025. A 'DELETE COMMENT' button is at the bottom right.

Figure 3.9.3: View report detail

- *Function detail:*
 - 1. User goes to the report detail page.
 - 2. The system fetches the data from the server.
 - 3. The system displays detailed data related to a report to the user

3.9.4 Update report statuses

- *Function trigger:* User clicks on the “Resolve” button dropdown.
- *Function description:* Allow the Administrator to resolve the reports sent by learners.
- *Screen layout:*

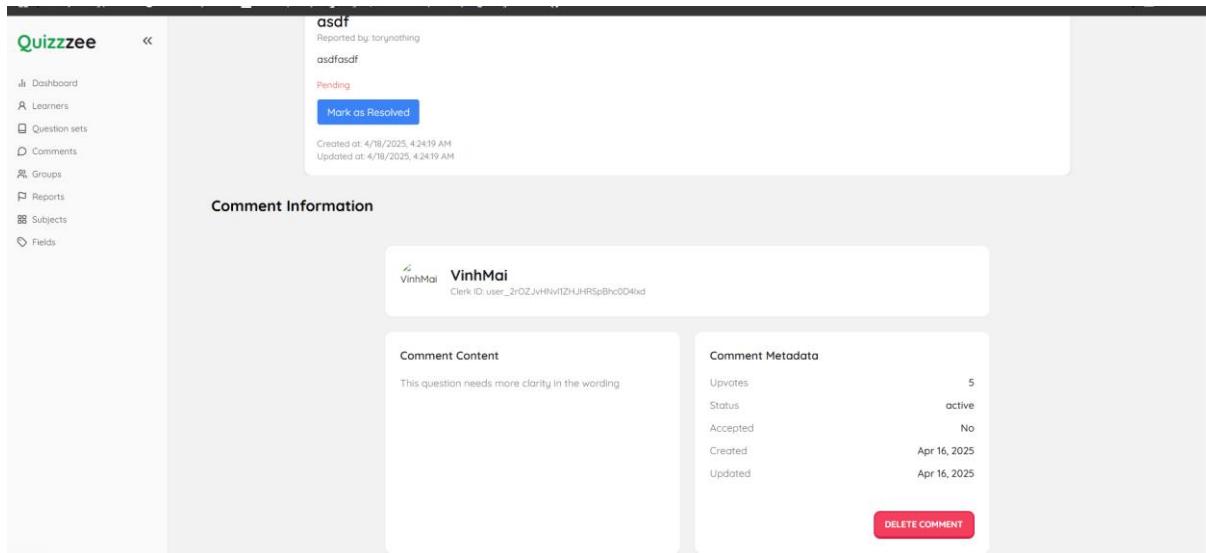


Figure 3.9.4: Update report statuses

- *Function detail:*
 - 1. User clicks on the “Resolve” button dropdown.
 - 2. User clicks on an option from the dropdown.
 - 3. The system displays a confirmed modal.
 - 4. User clicks on the “Resolve” button.
 - 5. The system sends the request to the server.
 - 6. The system displays a message to notify the user the report has been resolved.

3.10 Field management

3.10.1 View fields

- *Function trigger:* User goes to the explore page.
- *Function description:* Allow users to view the current fields inside the system.
- *Screen layout:*

The screenshot shows a user interface for an educational platform. At the top, there is a navigation bar with icons for HOME, EXPLORE, a search bar labeled 'Search with keyword', and user account information. Below the navigation bar, the title 'Explore' is displayed, followed by the sub-instruction 'Categorize your goals by subject, allowing you to focus on specific areas'. The main content area is divided into several cards, each representing a subject category:

- Biology**: Study of living organisms, their structure, function, growth, and evolution.
 - General Biology
 - Molecular Biology
 - Ecology
- Mathematics**: Study of numbers, quantities, shapes, and patterns.
 - Calculus
 - Linear Algebra
 - Statistics
- Physics**: Study of matter, energy, and the fundamental forces of nature.
 - Classical Mechanics
 - Quantum Physics
 - Electromagnetism
- Chemistry**: Study of substances, their properties, structure, and the changes they undergo.
 - Organic Chemistry
 - Inorganic Chemistry
 - Biochemistry
- History**
- Geography**

Figure 3.10.1: View fields

- *Function detail:*
 - 1. User goes to the explore page.
 - 2. The system fetches the data from the server.
 - 3. The system displays a list of created fields to the user.

3.10.2 View field detail

- *Function trigger:* User goes to the field detail page.
- *Function description:* Allow the Administrator to view the details of created fields.
- *Screen layout:*

The screenshot shows the Quizzzee application interface. On the left, there is a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. The 'Fields' link is highlighted. The main content area displays a card for the 'Biology' field. The card has a green header with the subject name and its Vietnamese translation 'Sinh học'. Below the header are two buttons: 'Update' (blue) and 'Delete' (red). The card is divided into two sections: 'English Description' and 'Vietnamese Description'. The English description states: 'Study of living organisms, their structure, function, growth, and evolution'. The Vietnamese description states: 'Nghiên cứu về sinh vật, cấu trúc, chức năng, sự phát triển và tiến hóa của chúng'. At the bottom of the card, there is a section titled 'Subjects' containing three sub-sections: 'General Biology' (Sinh học đại cương), 'Molecular Biology' (Sinh học phân tử), and 'Ecology' (Sinh thái học). Each sub-section includes a brief description of its focus.

Figure 3.10.2: View field detail

- *Function detail:*
 - 1. User goes to the field detail page.
 - 2. The system fetches the data from the server.
 - 3. The system displays detailed data related to a field to the user

3.10.3 Create fields

- *Function trigger:* User clicks on the “Create” button from the fields page.
- *Function description:* Allow the Administrator to create new fields for the system, these will be used for learners to easily find related subjects
- *Screen layout:*

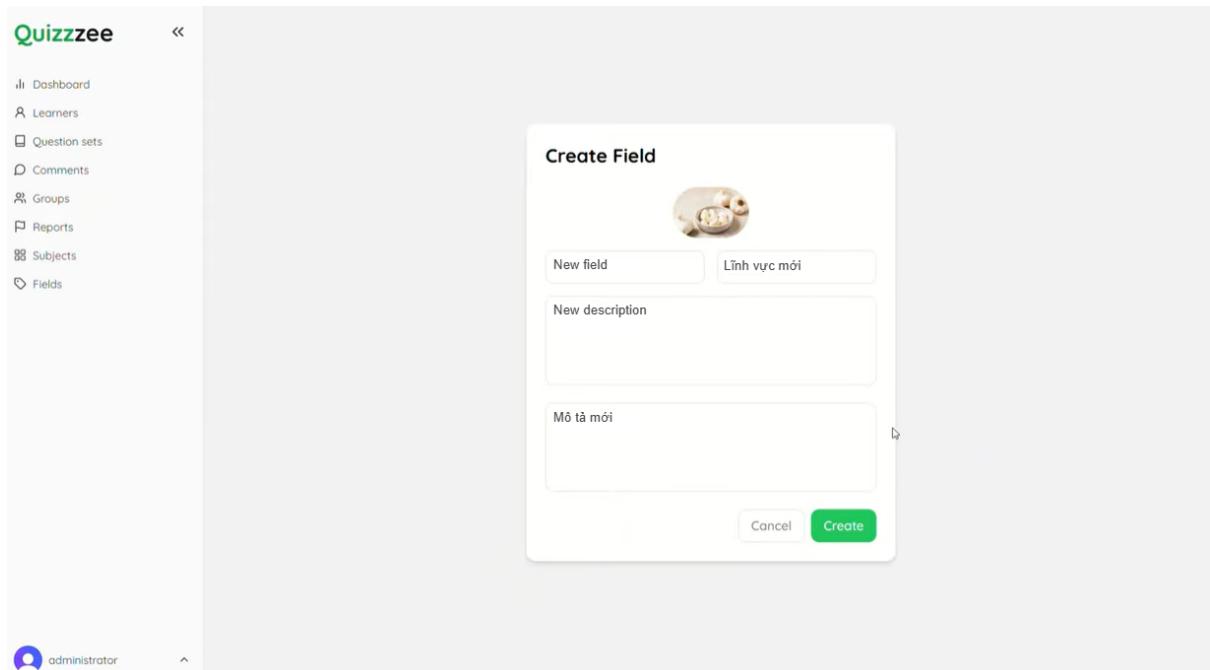


Figure 3.10.3: Create fields

- *Function detail:*
 - 1. User clicks on the “Create” button from the fields page.
 - 2. The system redirects the user to the create field page.
 - 3. User enters all the necessary fields.
 - 4. User clicks on the “Create” button.
 - 5. The system sends the data to the server.
 - 6. The system redirects the user to the fields page.
 - 7. The system updates the fields list to the user.

3.10.4 Update fields

- *Function trigger:* User clicks on the “Edit” button on the fields table row.
- *Function description:* Allow the Administrator to update the created fields.
- *Screen layout:*

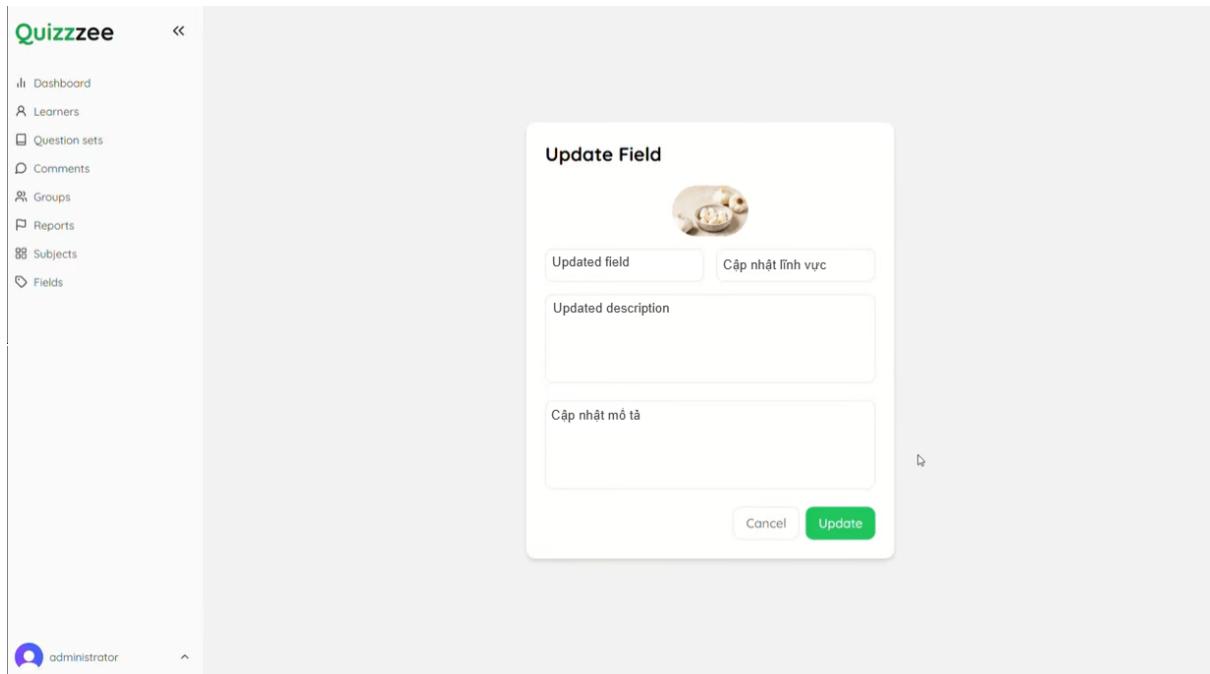


Figure 3.10.4: Update fields

- *Function detail:*
 - 1. User clicks on the “Edit” button on the fields table row..
 - 2. The system redirects the user to the update field page.
 - 3. The system fetches the related data from the server.
 - 4. The system displays the data to the user.
 - 5. User changes their desired data.
 - 6. User clicks the “Update” button.
 - 7. The system sends the modified data to the server.
 - 8. The system redirects the user to the fields page.
 - 9. The system updates the data to the user.

3.10.5 Delete fields

- *Function trigger:* user clicks on the delete button on the fields table row
- *Function description:* Allow the Administrator to delete the created fields
- *Screen layout:*

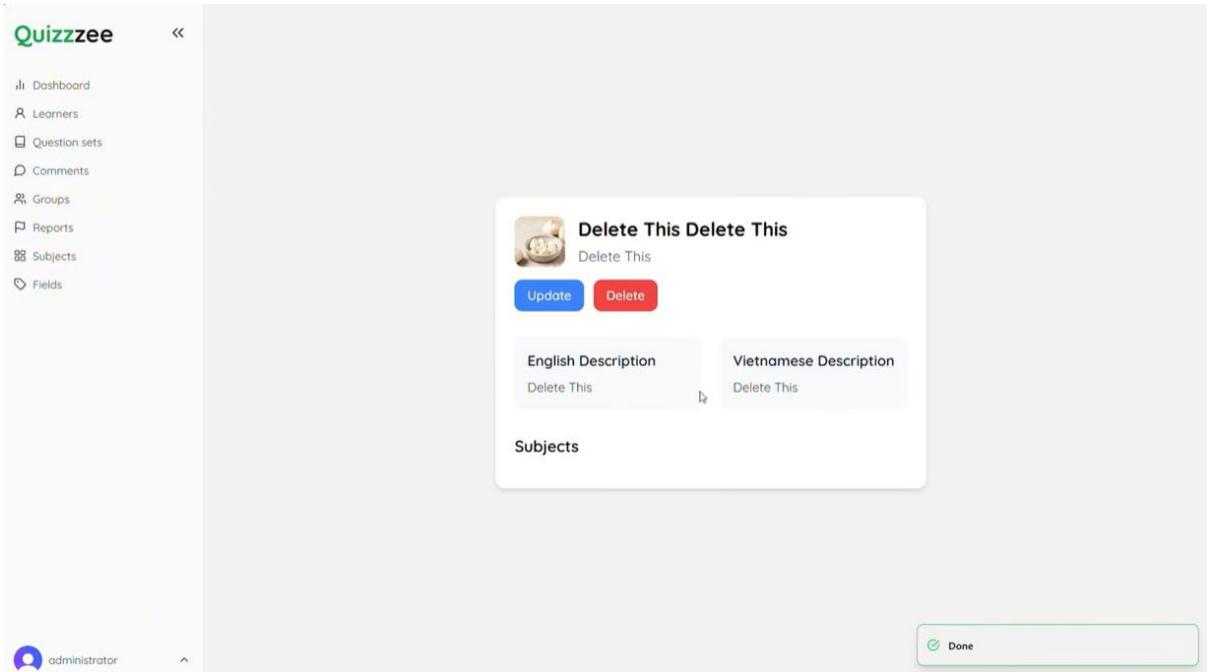


Figure 3.10.5: Delete fields

- *Function detail:*
 - 1. User clicks on the delete button on the fields table row.
 - 2. The system shows the modal to confirm user action.
 - 3. User clicks on the “Delete” button.
 - 4. The system sends the request to the server.
 - 5. The system displayed an updated fields table to the user.

3.11 Subject management

3.11.1 View subjects

- *Function trigger:* User goes to the explore page.
- *Function description:* Allow users to view the current subjects on the system.
- *Screen layout:*

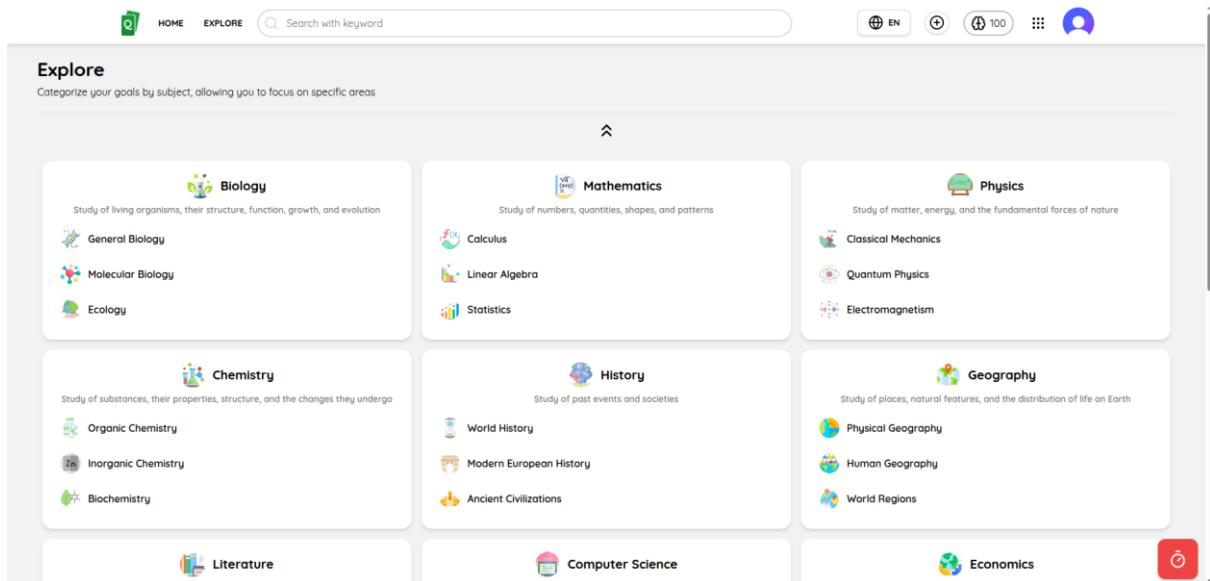


Figure 3.11.1: View subjects

- *Function detail:*
 - 1. User goes to the explore page.
 - 2. The system fetches the data from the server.
 - 3. The system displays a list of created subjects to the user.

3.11.2 View subject detail

- *Function trigger:* User goes to the subject detail page.
- *Function description:* Allow the Administrator to view the details of a created subject.
- *Screen layout:*

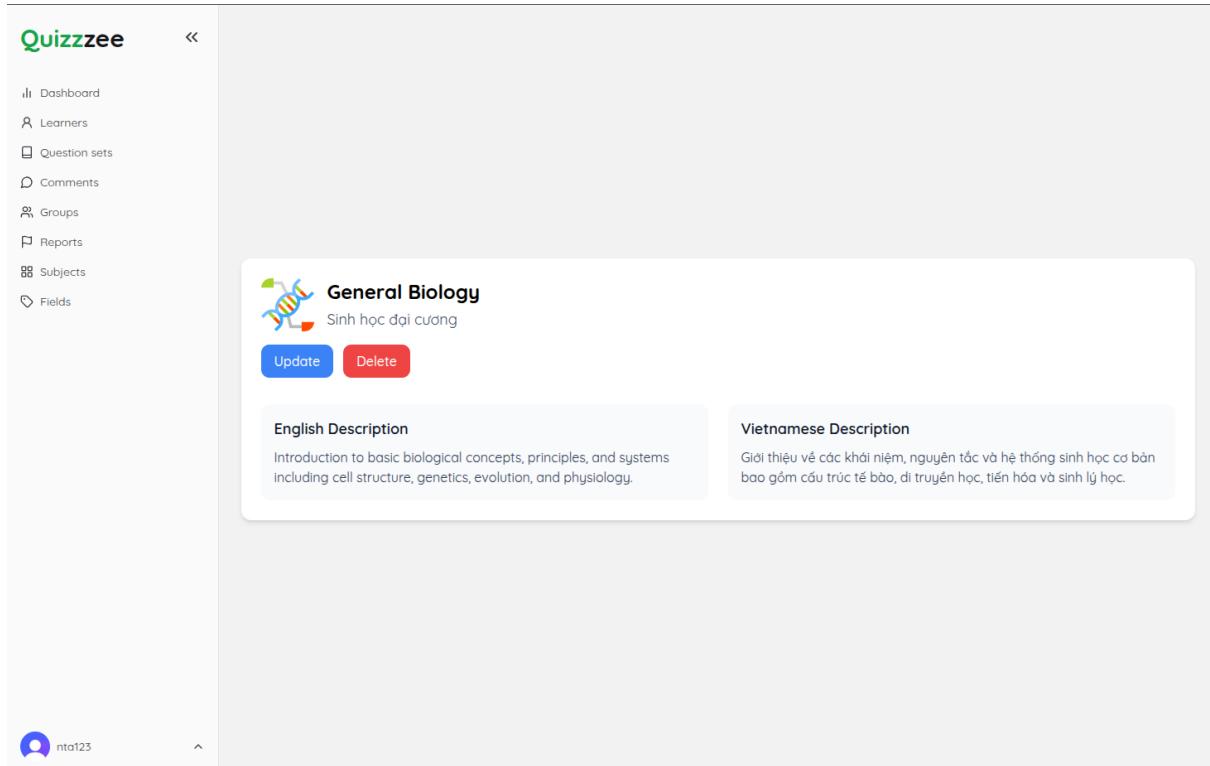


Figure 3.11.2: View subject detail

- *Function detail:*
 - 1. User goes to the subject detail page.
 - 2. The system fetches the data from the server.
 - 3. The system displays detailed data related to a subject to the user

3.11.3 Create subjects

- *Function trigger:* User clicks on the “Create” button from the subjects page.
- *Function description:* Allow the Administrator to create new subjects for the system, these will be used for learners to categories their question sets.
- *Screen layout:*

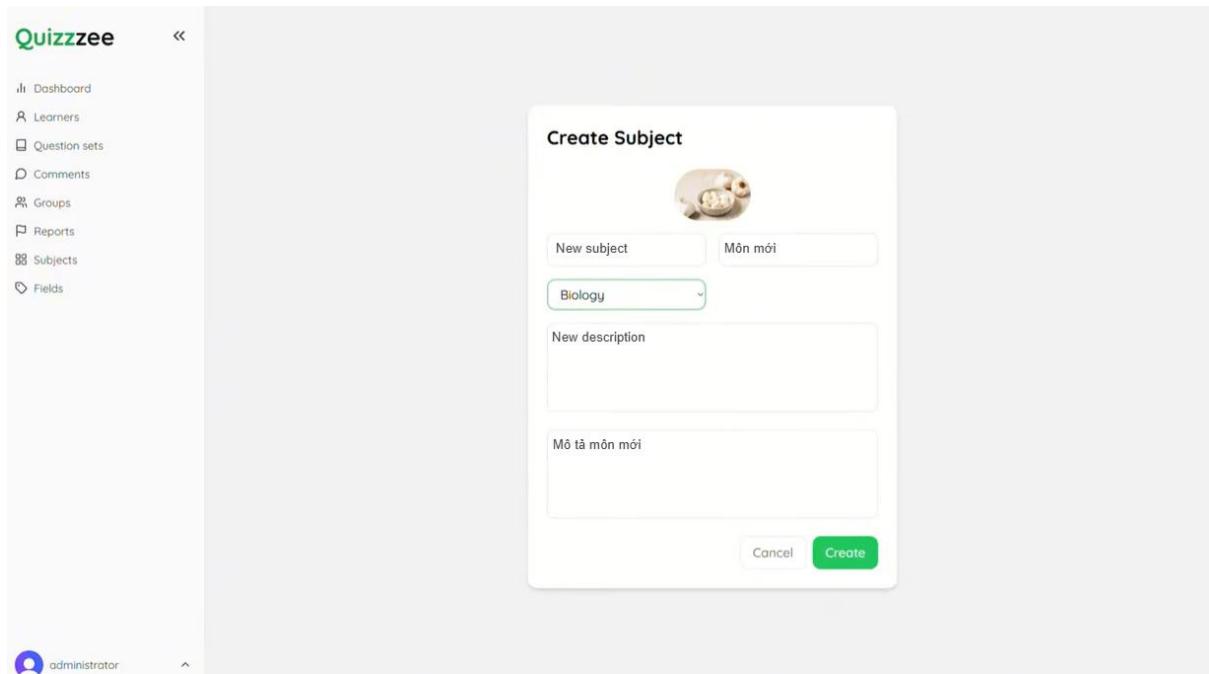


Figure 3.11.2: Create subjects

- *Function detail:*

1. User clicks on the “Create” button from the subject’s page.
2. The system redirects the user to the create subject’s page.
3. User enters all the necessary fields.
4. User clicks on the “Create” button.
5. The system sends the data to the server.
6. The system redirects the user to the subject’s page.
7. The system updates the fields list to the user.

3.11.4 Update subjects

- *Function trigger:* User clicks on the “Edit” button on the subjects table row.
- *Function description:* Allow the Administrator to update the created subjects.
- *Screen layout:*

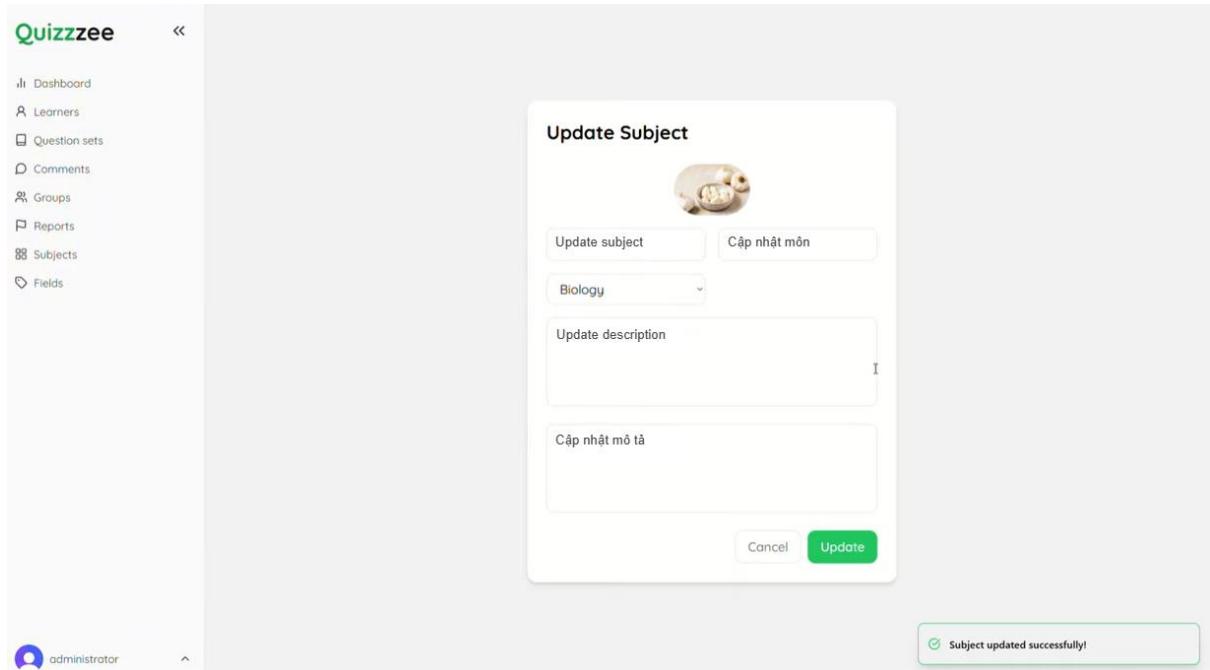


Figure 3.11.3: Update subjects

- *Function detail:*
 - 1. User clicks on the “Edit” button on the subjects table row..
 - 2. The system redirects the user to the update subject page.
 - 3. The system fetches the related data from the server.
 - 4. The system displays the data to the user.
 - 5. User changes their desired data.
 - 6. User clicks the “Update” button.
 - 7. The system sends the modified data to the server.
 - 7. The system redirects the user to the subjects page.
 - 8. The system updates the data to the user.

3.11.5 Delete subjects

- *Function trigger:* User clicks on the delete button on the subject table row
- *Function description:* Allow the Administrator to delete the created subjects
- *Screen layout:*

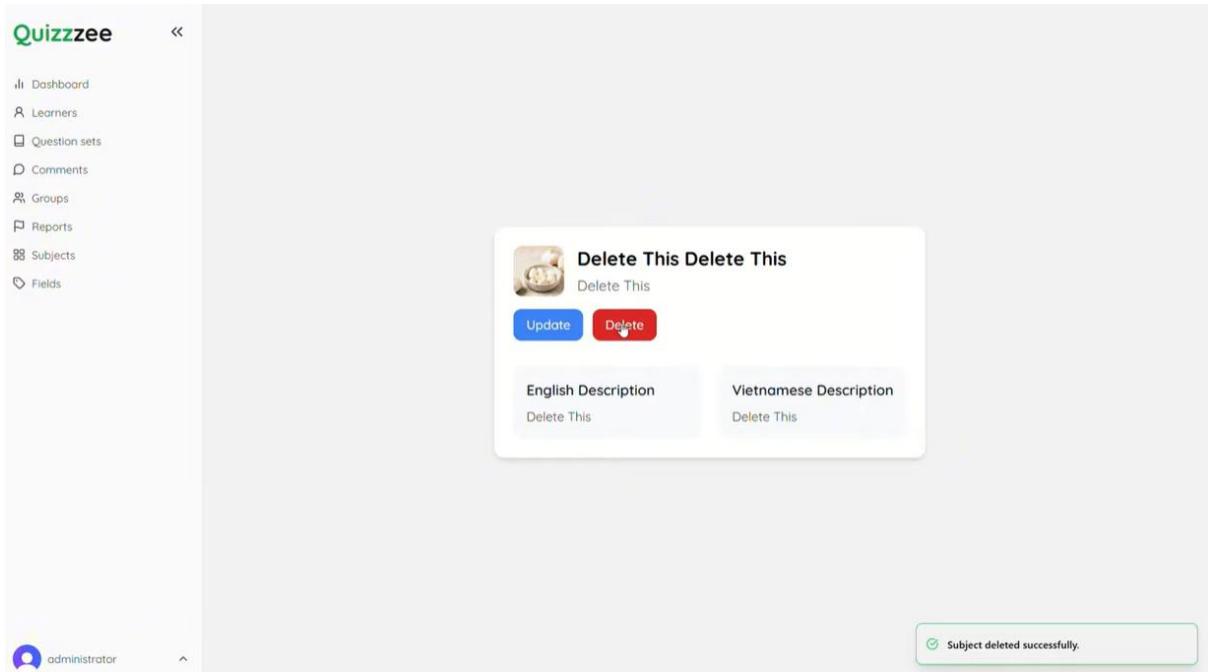


Figure 3.11.4: Delete subjects

- *Function detail:*
 - 1. User clicks on the delete button on the subject table row.
 - 2. The system shows the modal to confirm user action.
 - 3. User clicks on the “Delete” button.
 - 4. The system sends the request to the server.
 - 5. The system displayed an updated subjects table to the user.

3.12 Following management

3.12.1 Follow account

- *Function trigger:* User clicks on the “Follow” button
- *Function description:* Allow users to follow other user accounts.
- *Screen layout:*

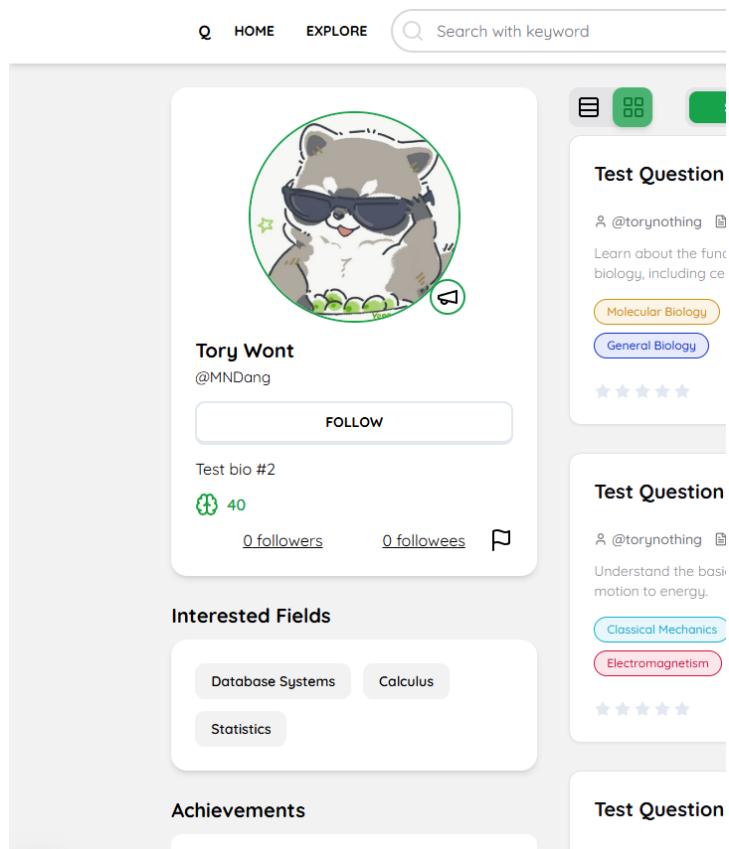


Figure 3.12.1: Follow account

- *Function detail:*
 - 1. User clicks on the “Follow” button.
 - 2. The system requests the server to update the status.
 - 3. The system changes the “Follow” button title to “Following”.

3.12.2 View follow detail

- *Function trigger:* User click on the followings or follower's number
- *Function description:* Allow users to view the following and follower details of the learner.
- *Screen layout:*

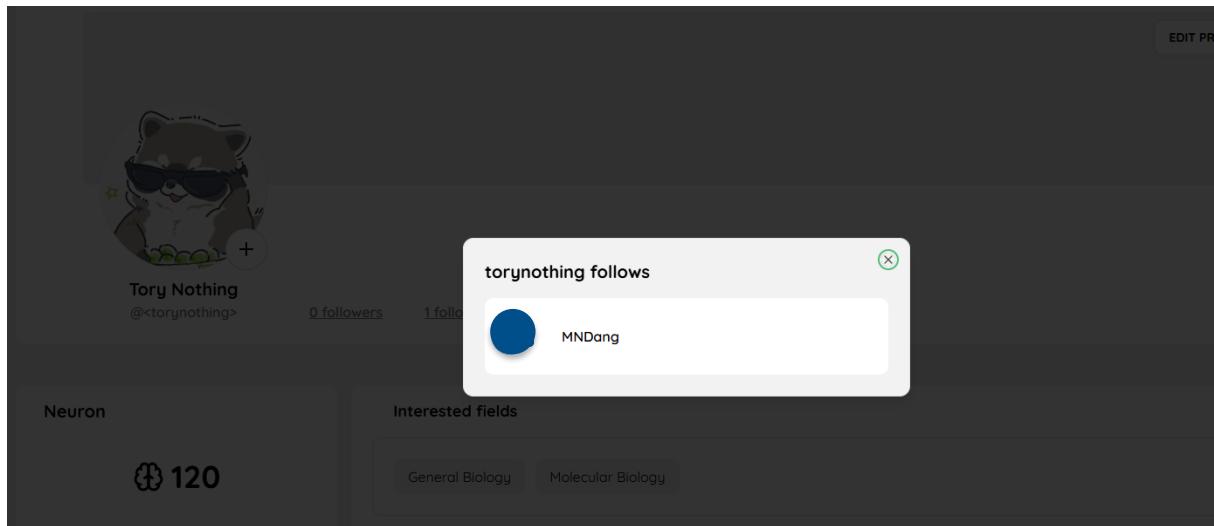


Figure 3.12.2: View follow detail

- *Function detail:*
 - 1. User clicks on the followings or follower's number.
 - 2. The system requests the data from the server.
 - 3. The system displays the follow detail modal

3.13 Question set progress management

3.13.1 Mark questions

- *Function trigger:* User clicks on the star icon from the question card inside the question set detail page.
- *Function description:* Allow users to mark questions inside a question set, users later can filter out questions based on these mark questions.
- *Screen layout:*

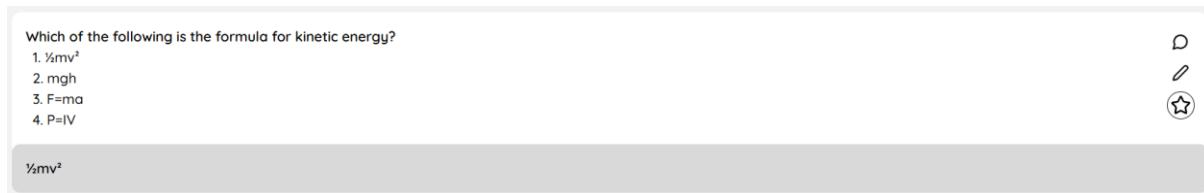


Figure 3.13.1: Mark questions

- *Function detail:*
 - 1. User scrolls to the question they want to mark.
 - 2. User clicks on the star icon.
 - 3. The system sends the request to the server.
 - 4. The system updates the look of the star icon of the current question card.

3.13.2 Filter questions

- *Function trigger:* User inputs keywords into the search bar inside the question set detail page.
- *Function description:* Allow users to filter questions inside a question set using keywords or with available filter options.
- *Screen layout:*

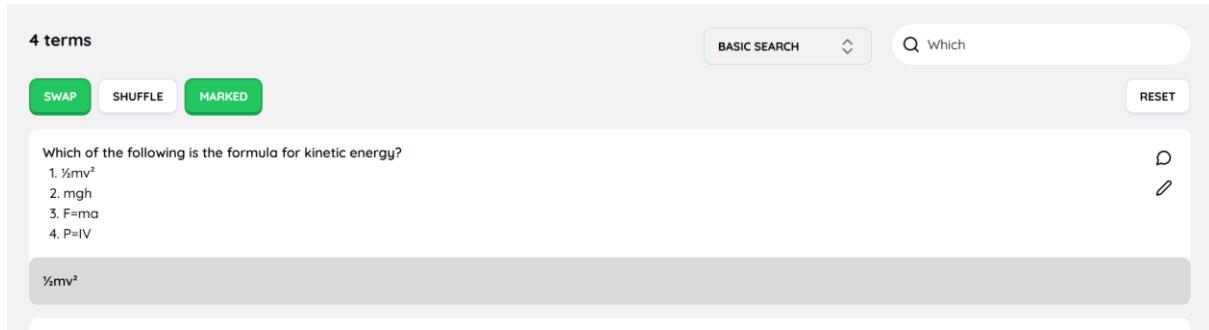


Figure 3.13.2: Filter questions

- *Function detail:*
 - 1. User inputs keywords into the search bar inside the question set detail page.
 - 2. The system fetches the data from the server.
 - 3. The system displays a list of questions that match the user filter.

3.14 Comment management

3.14.1 View comments

- *Function trigger:* User goes to the question set detail page.
- *Function description:* Allow users to view comments created by other learners inside a question set.
- *Screen layout:*

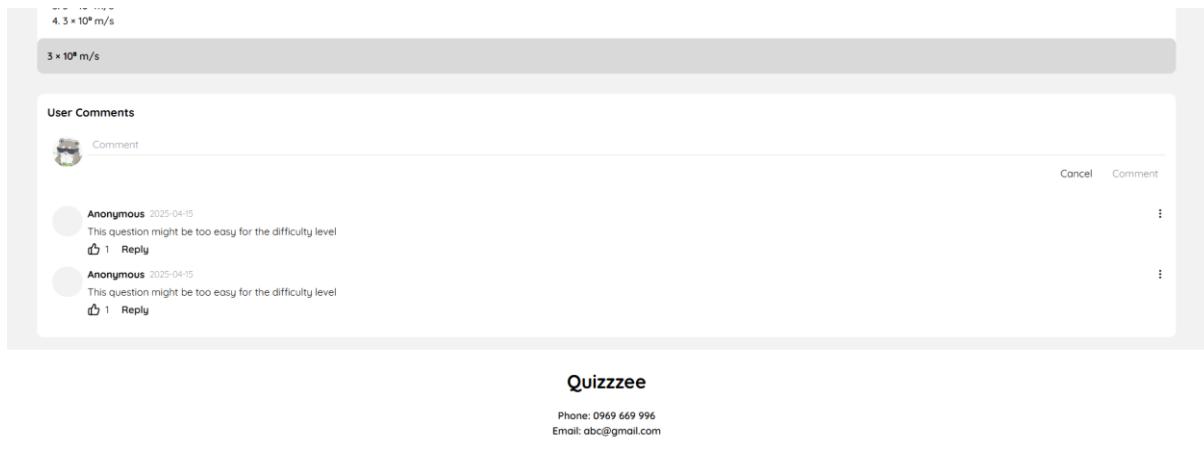


Figure 3.14.1: View comments

- *Function detail:*
 - 1. User goes to the question set detail page.
 - 2. User scrolls down to the comment section.
 - 3. The system fetches the data from the server.
 - 4. The system displays a section of comments to the user.

3.14.2 Filter comments

- *Function trigger:* User chooses a filter option from the comment section.
- *Function description:* Allow users to filter comments base on date created
- *Screen layout:*

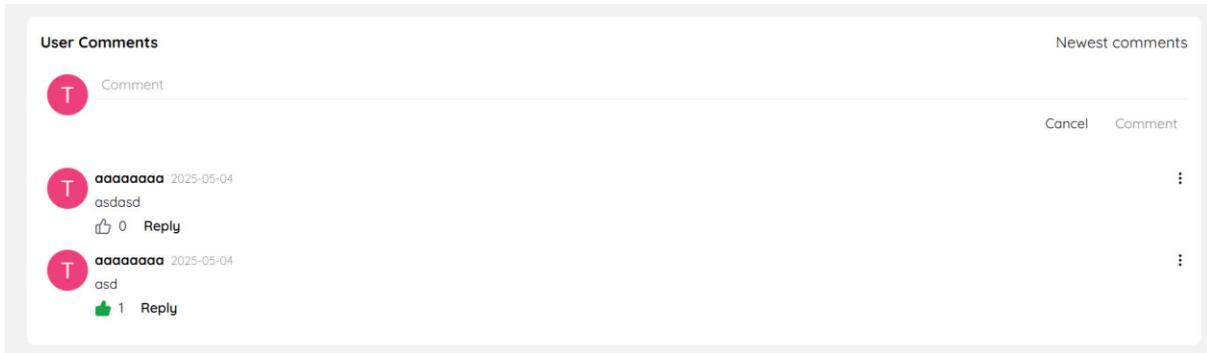
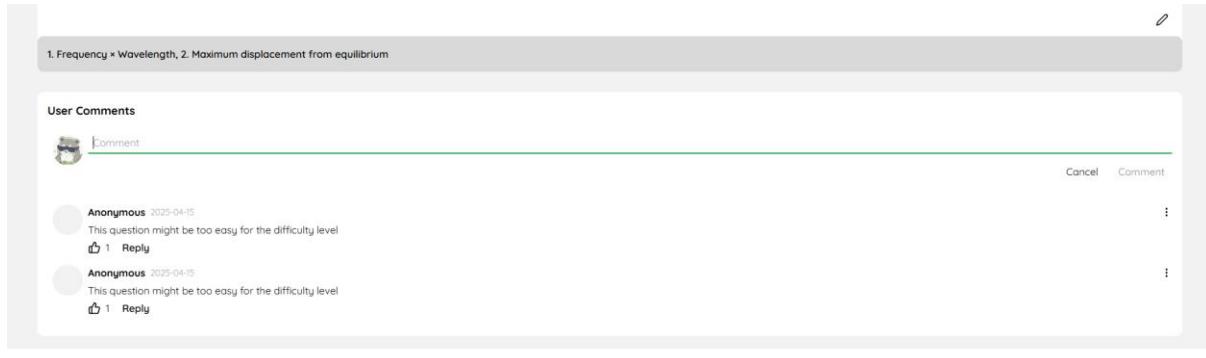


Figure 3.14.2: Filter comments

- *Function detail:*
 - 1. User chooses a filter option from the comment section
 - 2. The system fetches the data from the server.
 - 3. The system displays a section of related comments to the user.

3.14.3 Create comments

- *Function trigger:* User inputs a comment.
- *Function description:* Allow users to create comments inside a question set.
- *Screen layout:*



Quizzzee

Figure 3.14.3: Create comments

- *Function detail:*
 - 1. User inputs a comment.
 - 2. User clicks on the “Comment” button.
 - 3. The system sends the data to the server.
 - 4. The system displays the comment at the top of the comment section.

3.14.4 Mark question comments

- *Function trigger:* User hovers and clicks on the check mark icon inside a question comment
- *Function description:* Allow the author of the question set to mark a question comment, which means the question comment is approved by the author.
- *Screen layout:*

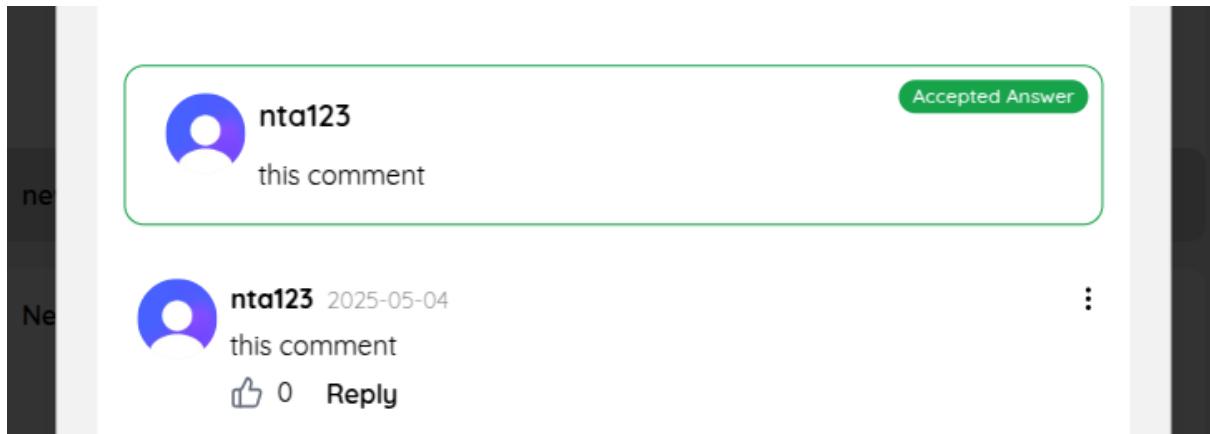


Figure 3.14.4: Mark question comments

- *Function detail:*
 - 1. User hovers and clicks on the check mark icon inside a question comment.
 - 2. The system sends the request to the server.
 - 3. The system displays a checkmark next to the marked question comment.
 - 4. The system displays a message to notify the user the comment has been marked.

3.14.5 Update comments

- *Function trigger:* User clicks on the edit option inside the action dropdown of the comments.
- *Function description:* Allow users to update their created comments.
- *Screen layout:*



Figure 3.14.5: Update comments

- *Function detail:*
 - 1. User clicks on the edit option inside the action dropdown of the comments.
 - 2. User edits the comment.
 - 3. User clicks on the “Edit” button.
 - 4. The system sends the data to the server.
 - 5. The system displays the updated comment to the user.

3.14.6 Delete comments

- *Function trigger:* User clicks on the delete option inside the action dropdown of the comments.
- *Function description:* Allow users to delete their created comment
- *Screen layout:*

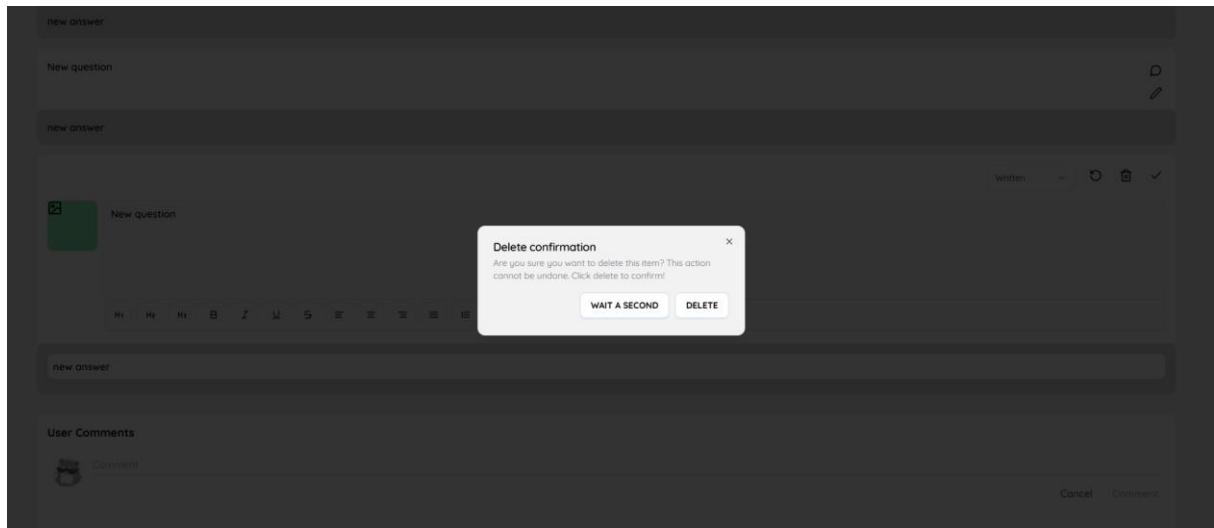


Figure 3.14.6: Delete comments

- *Function detail:*
 - 1. User clicks on the delete option inside the action dropdown of the comments.
 - 2. The system displays the confirm modal.
 - 3. User clicks on the “Delete” button.
 - 4. The system sends the data to the server.
 - 5. The system removes the comment from the comment section.

3.14.7 Reply comments

- *Function trigger:* User clicks on the “Reply” button under a comment.
- *Function description:* Allow users to create replies to a comment inside a question set.
- *Screen layout:*



Figure 3.14.7: Reply comments

- *Function detail:*
 - 1. User clicks on the “Reply” button under a comment.
 - 2. User enters the reply comment.
 - 3. User clicks on the “Reply” button.
 - 4. The system sends the data to the server.
 - 5. The system displays the replied comment under the initial comment.

3.14.8 Upvote comments

- *Function trigger:* User clicks on the thumb up icon under a comment.
- *Function description:* Allow user to upvote a comment inside question set
- *Screen layout:*

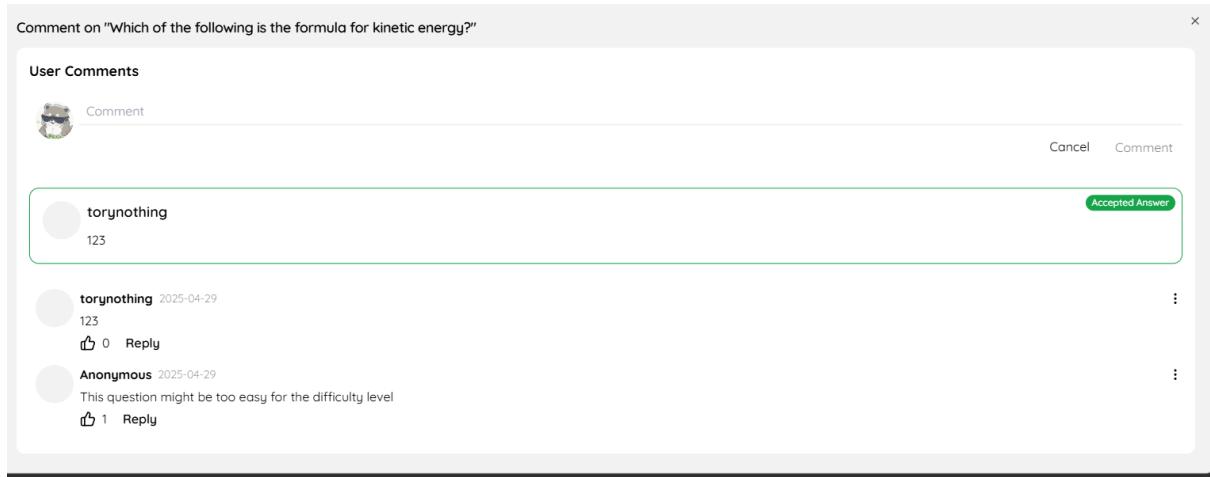


Figure 3.14.8: Upvote comments

- *Function detail:*
 - 1. User clicks on the thumb up icon under a comment.
 - 2. The system sends the request to the server.
 - 3. The system displays the updated number of upvotes to the user.

3.15 Learn mode Interaction

3.15.1 Use learn mode

- *Function trigger:* User clicks on the “Learn” button inside the question set detail page
- *Function description:* Allow users to learn a question set and have the system tracking their progress.
- *Screen layout:*

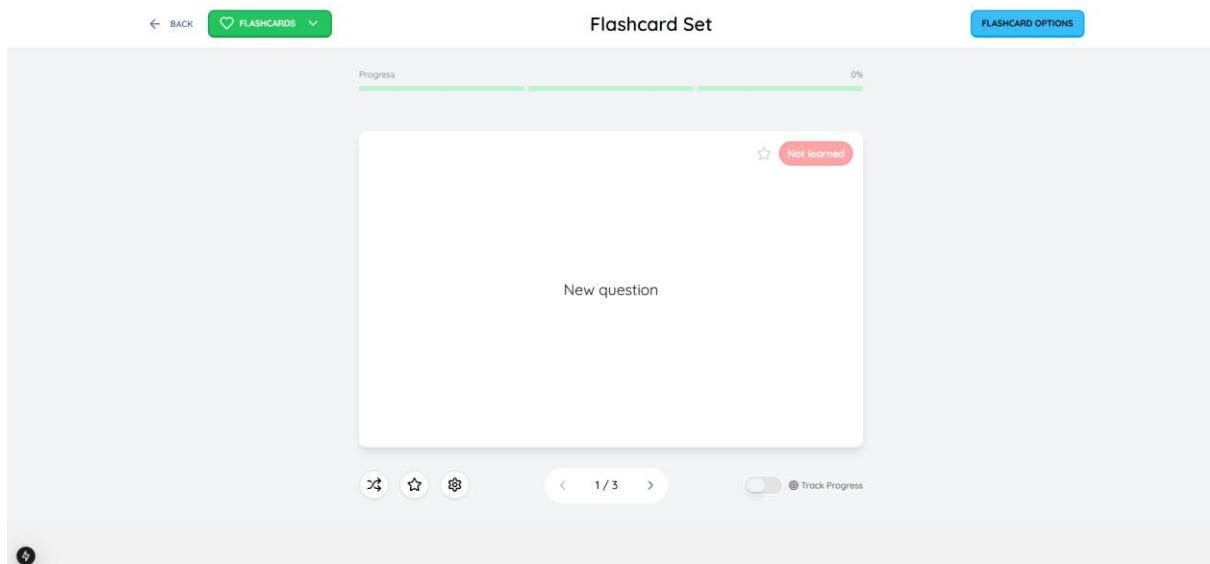


Figure 3.15.1: Use learn mode

- *Function detail:*
 - 1. User clicks on the “Learn” button inside the question set detail page.
 - 2. The system redirects the user to the learn page.
 - 3. The system fetches the data from the server.
 - 4. The system displays the questions as flash cards for the user.

3.15.2 Use matching mode

- *Function trigger:* User clicks on the “Matching” button inside the question set detail page.
- *Function description:* Allow users to learn a question set by playing matching game,
- *Screen layout:*

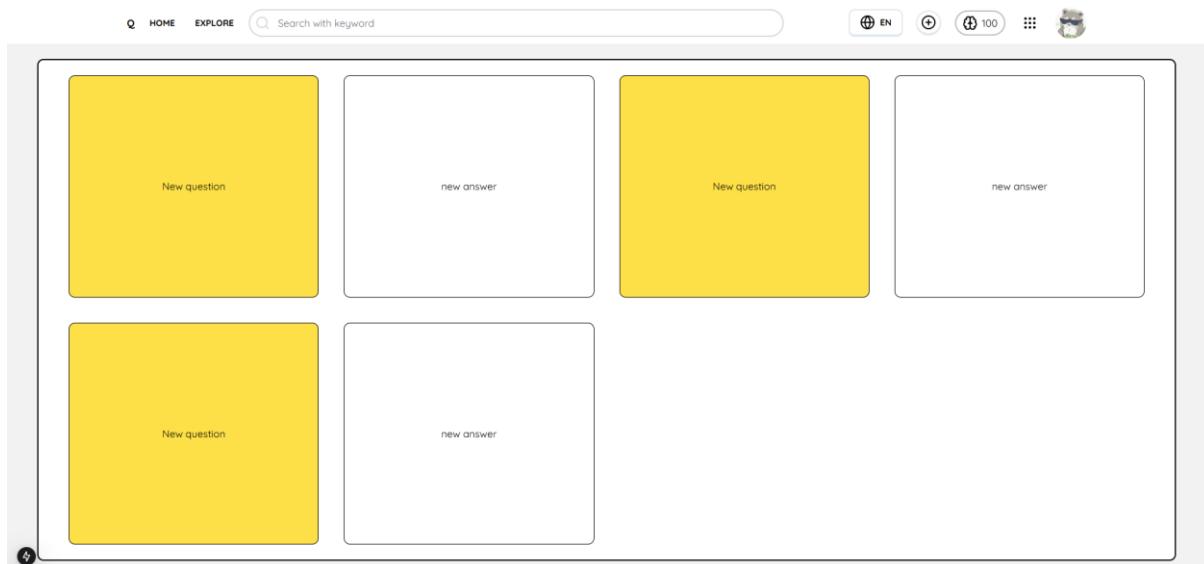


Figure 3.15.2: Use matching mode

- *Function detail:*
 - 1. User clicks on the “Matching” button inside the question set detail page.
 - 2. The system redirects the user to the matching page.
 - 3. The system fetches the data from the server.
 - 4. The system displays the questions and answers as separate cards for the user.

3.16 Quiz mode interaction

3.16.1 Use quiz mode

- *Function trigger:* User clicks on the “Quiz” button inside the question set detail page
- *Function description:* Allow users to learn a question set by doing quizzes, the system will then calculate the final score and feedback for the user.
- *Screen layout:*

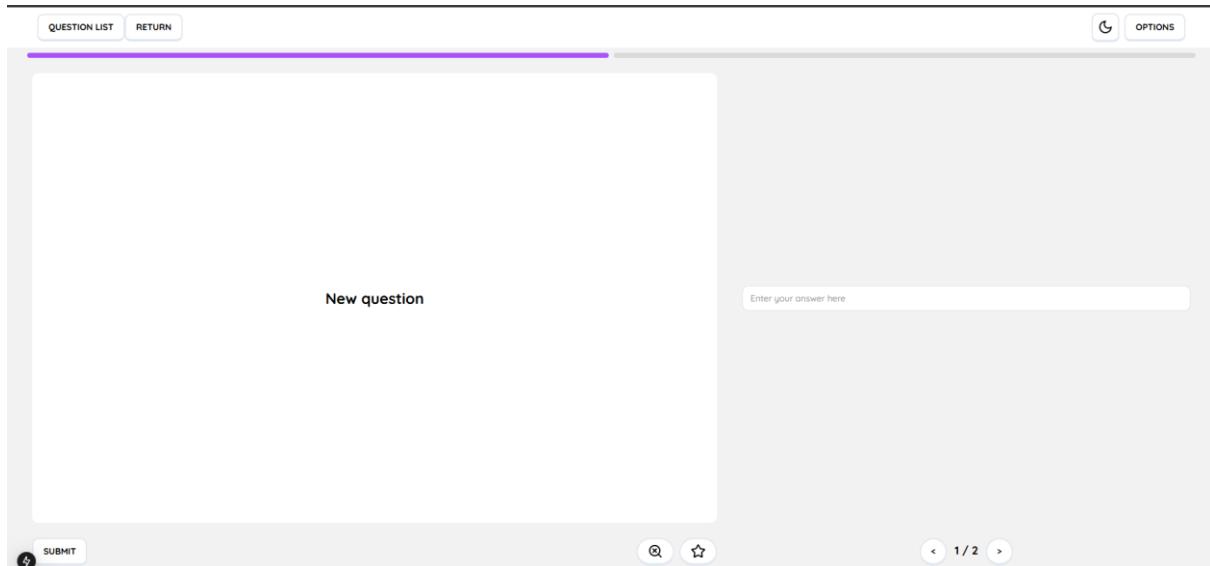


Figure 3.16.1: Use quiz mode

- *Function detail:*
 - 1. User clicks on the “Quiz” button inside the question set detail page.
 - 2. The system redirects the user to the quiz page.
 - 3. The system displays the quiz option modal.
 - 4. User customizes the option and clicks the start button.
 - 5. The system fetches the data from the server.
 - 6. The system displays the question as quizzes.
 - 7. User chooses/ inputs the answer and clicks the “Submit” button.
 - 8. The system sends the data to the server.
 - 9. The system displays the total score to the users.

3.16.2 Revisit quiz attempts

- *Function trigger:* User clicks on the “Revisit” button inside quiz result card
- *Function description:* Allow users to revisit their previous quiz attempts, which displays what users have answered and total score.
- *Screen layout:*

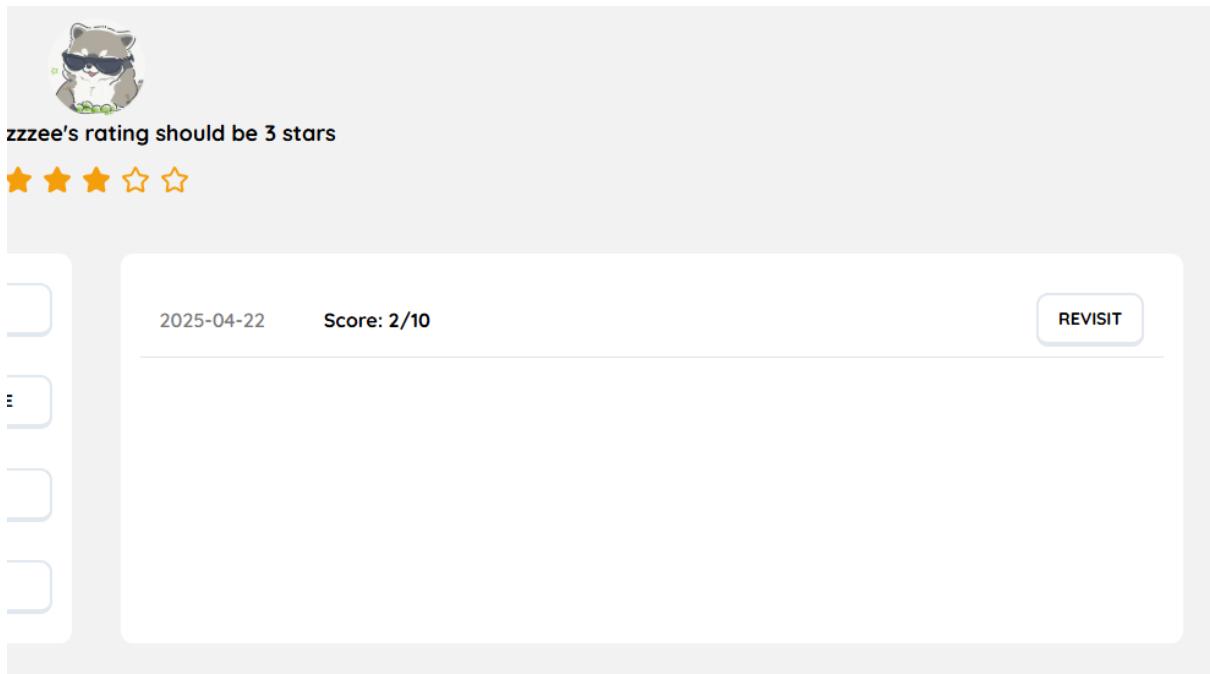


Figure 3.16.2: Revisit quiz attempts

- *Function detail:*
 - 1. User clicks on the “Revisit” button inside the quiz result card.
 - 2. The system redirects the user to the quiz attempt page.
 - 3. The system fetches the data from the server.
 - 4. The system displays the details of previous quiz attempts to the user.

3.16.3 View quiz statistics

- *Function trigger:* User goes to the question set detail page.
- *Function description:* Allow users to view the overall statistics of their question set progress and previous quiz attempts.
- *Screen layout:*

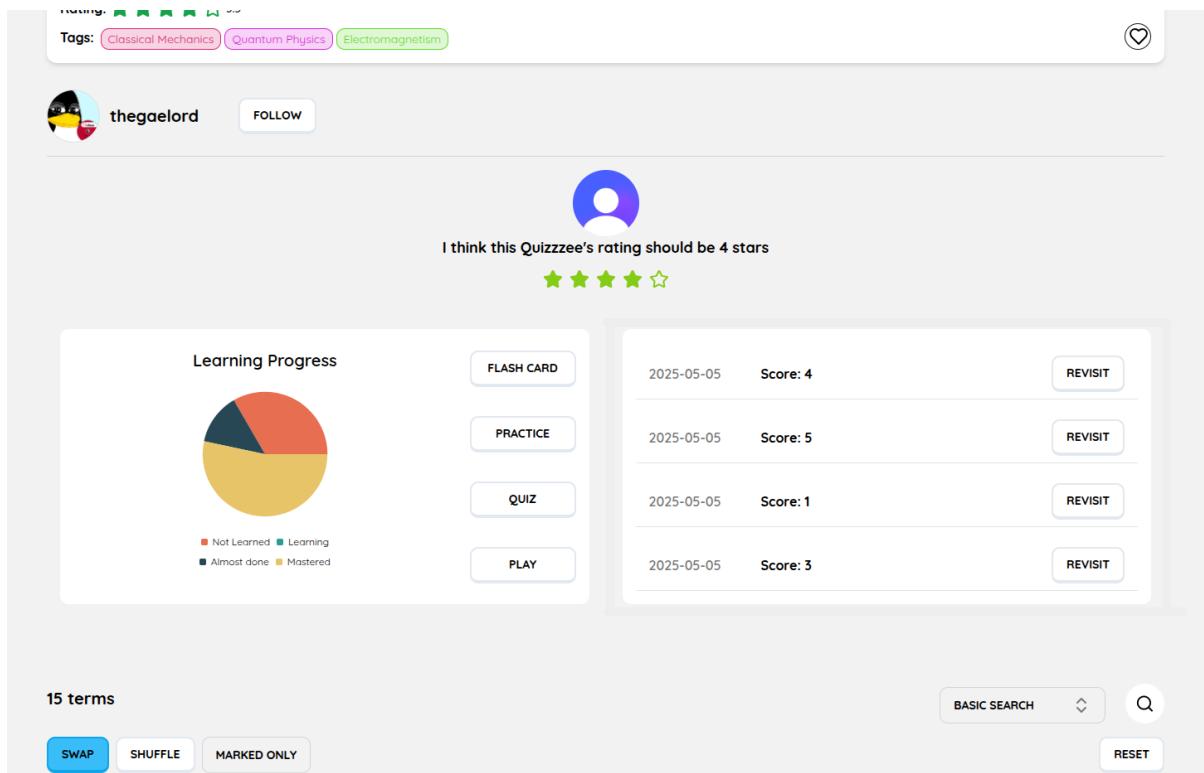


Figure 3.17.3: View quiz statistics

- *Function detail:*
 - 1. User goes to the question set detail page.
 - 2. The system fetches data from the server.
 - 3. The system displays the statistics data to the user.

3.17 Statistics interaction

3.17.1 View statistics

- *Function trigger:* User goes to the Dashboard page
- *Function description:* Allow the Administrator to view the system's overall statistics.
- *Screen layout:*

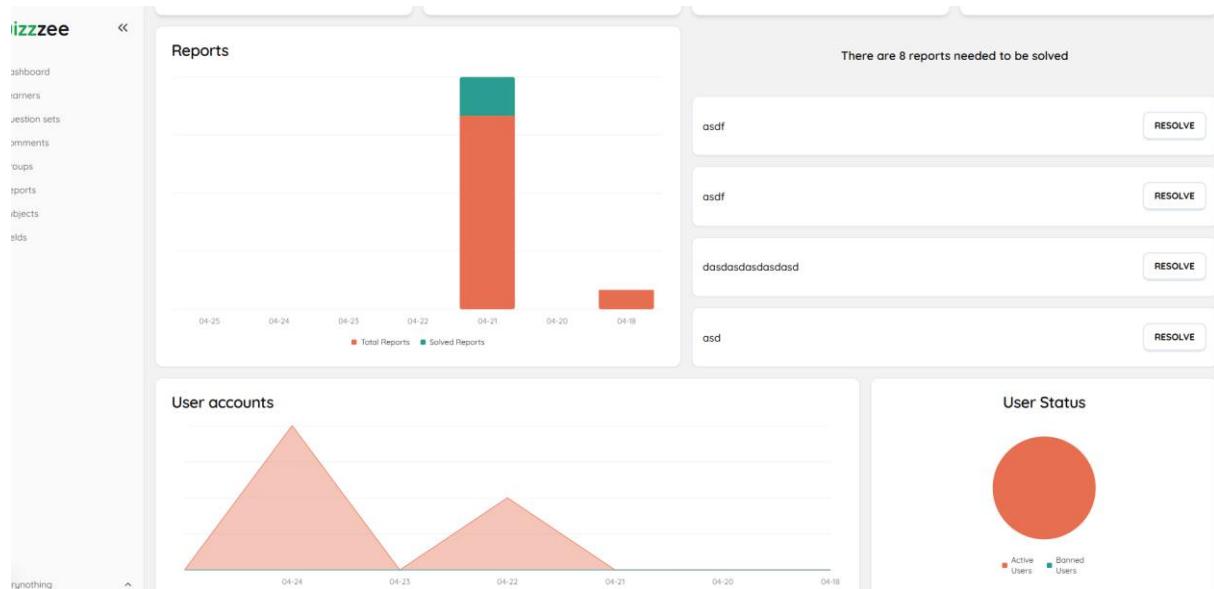


Figure 3.17.1: View statistics

- *Function detail:*
 - 1. User goes to the Dashboard page.
 - 2. The system fetches the data from the server.
 - 3. The system displays related data to the user.

3.17.2 Filter statistics

- *Function trigger:* Users search a keyword on the search bar inside the Dashboard page.
- *Function description:* Allow the Administrator to filter different system statistics either by using keywords or date range.
- *Screen layout:*

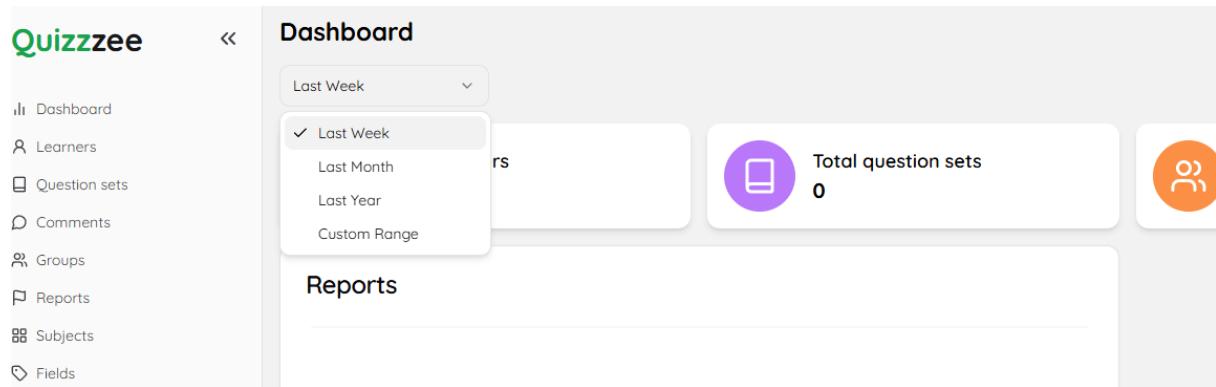


Figure 3.17.2: Filter statistics

- *Function detail:*
 - 1. User searches for a keyword on the search bar inside the Dashboard page.
 - 2. The system fetches the data from the server.
 - 3. The system displays the related data to the user.

3.18 Additional features

3.18.1 Use Pomodoro timer

- *Function trigger:* User clicks on the Pomodoro button.
- *Function description:* Allow users to keep track of their own working session using the Pomodoro method, including managing their own session duration.
- *Screen layout:*

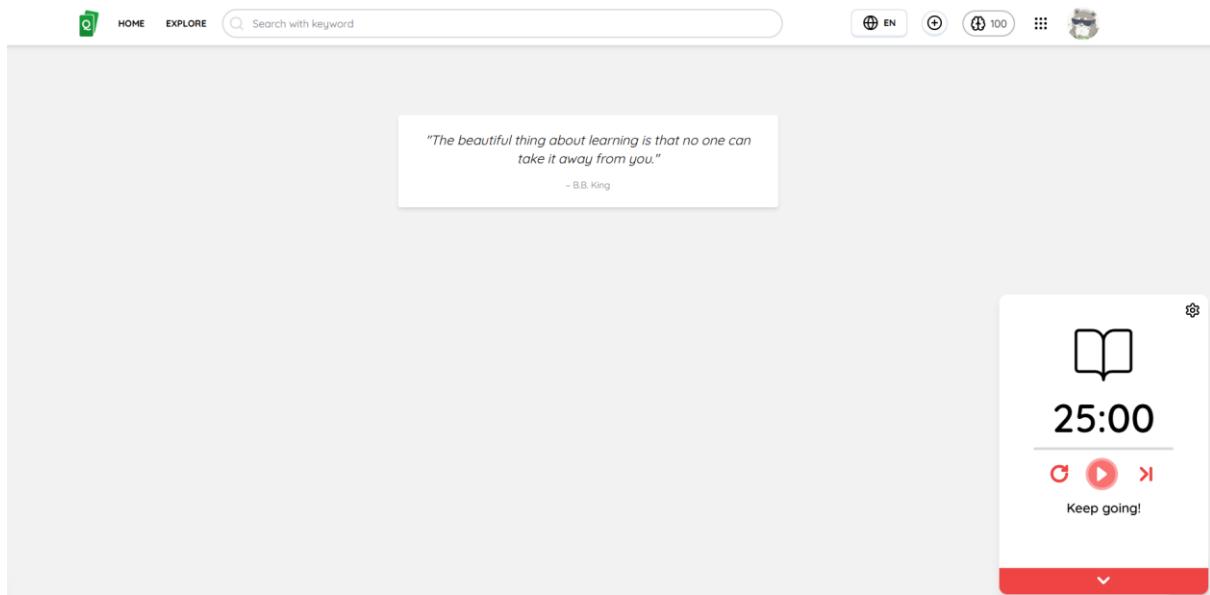


Figure 3.18.1: Use Pomodoro timer

- *Function detail:*
 - 1. User clicks on the Pomodoro button.
 - 2. User clicks on the Start button inside the Pomodoro overlay.
 - 3. The timer starts counting down.
 - 4. The timer displays a sound to notify the timer has reached 0.
 - 5. The timer changes the Start button to Pause button

3.18.2 View achievements

- *Function trigger:* User clicks on the “Achievements” button.
- *Function description:* Allow users to view available achievements in the system.
- *Screen layout:*

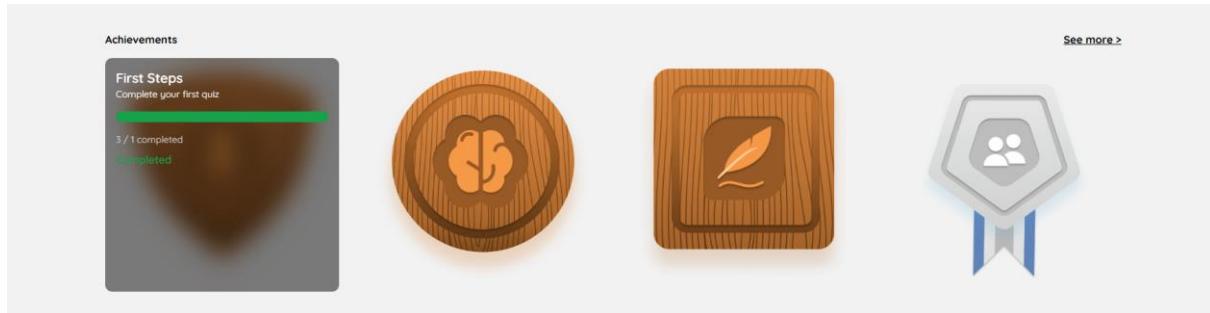


Figure 3.18.2: View achievements

- *Function detail:*
 - 1. User clicks on the “Achievements” button.
 - 2. The system requests all available achievements from the server.
 - 3. The system displays the achievements to the user.

3.18.3 View quote

- *Function trigger:* User goes to the home page
- *Function description:* Allow users to view famous quotes in the system.
- *Screen layout:*

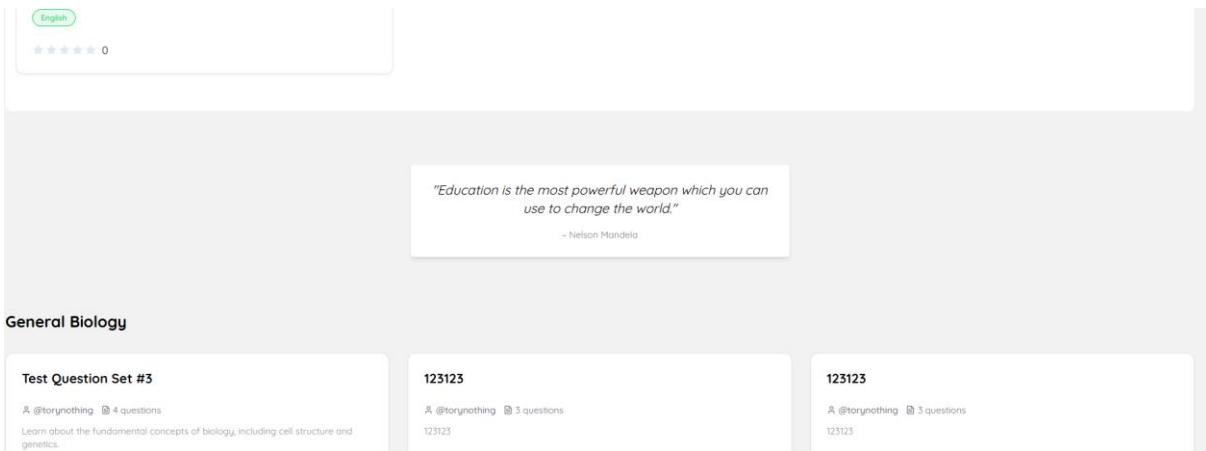


Figure 3.18.3: View quotes

- *Function detail:*
 - 1. User goes to the home page.
 - 2. The system gets the data from locals.
 - 3. The system displays the quote to the user.

4. Non-Functional Requirements

4.1 External Interfaces

4.1.1 Software Interfaces

- SI-01: The website should be compatible with the latest versions of major browsers like Chrome, Firefox, Safari or Edge.
- SI-02: The website should provide clear and quick responses to user action.
- SI-03: The website must load within 5 seconds to ensure quick and responsive user experience.
- SI-04: The error messages should be informative and helpful to the users.
- SI-05: The website should be responsive and adapt to different screen sizes and devices.

4.1.2 Hardware Interfaces

- HI-01: The website should be compatible with a wide range of hardware configurations, including different processors, memory capacities, and screen resolutions.
- HI-02: The website should function correctly on common input devices (e.g., mouse, keyboard, touch screen).
- HI-03: The website should provide clear instructions and feedback when interacting with peripheral devices.

4.1.3 Communication Interfaces

- CI-01: The frontend interface of the system communicates with the backend server through HTTP requests and HTTP responses to ensure smooth and efficient data exchange.
- CI-02: The system should handle network interruptions and communication failures gracefully.
- CI-03: Communication interfaces should protect sensitive data from unauthorized access.
- CI-04: Third-party service integrations should be secure, seamless and reliable.

4.2 Quality Attributes

4.2.1 Security

- User passwords must be strongly encrypted to safeguard personal data.
- Robust authentication and authorization must be implemented to restrict access to functions based on user roles.

4.2.2 Performance

- The system must maintain stability and performance under heavy user traffic.
- User interactions will have minimal response times for a smooth experience.
- The system database must be designed to be scalable and easily adapt to growth in users and data.

4.2.3 Reliability and availability

- The system must include features for recovering from failures and preventing data loss.
- The system must operate smoothly for 90% daily uptime to ensure consistent access.

4.2.4 User interface

- The user interface must be designed to be user-friendly, intuitive, and compatible across various devices and browsers.
- The user interface must be optimized to provide the best possible user experience.

4.2.5 Service and support

- Technical support and regular system updates are required to ensure stability and performance.
- Detailed documentation and user guides must be provided so that users can easily and effectively use the system.

5. Requirement Appendix

5.1 Business Rules

ID	Rule Definition
BR-01	Learners must be the group leader to modify group settings.
BR-02	Learners must follow other learners before they can unfollow them.
BR-03	Only group members can send invitations to non-group members.
BR-04	Only the author of the question set can mark other learners' comments.
BR-05	Group invitation only exists for 3 days before permanently removed
BR-06	User must use unique username or email to sign up or sign in to the system

BR-07	Learners' quiz attempts must be stored for statistics
BR-08	Administrator must sign in before using the system
BR-09	A question set must contain at least 3 questions
BR-10	User's password must be at least 8 characters long
BR-11	User's credential data must be valid
BR-12	Question comments with mark will be shown first inside each question
BR-13	Inactive or private data will not be displayed publicly
BR-14	Each question can only have one marked comment
BR-15	Private question sets can only be viewed by the author
BR-16	Data related to inactive question sets will not be shown to users
BR-17	The updated comments will only change the comment itself, the upvote, replies and author check mark will not be reset.
BR-18	A learner's list of favourited question sets will not be shown to other users
BR-19	Only the author or participated member can view the private groups
BR-20	Removed group members can still rejoin the group that removed them, but not the banned member.
BR-21	Goals with a deadline less than 1 day will be notified to the user.
BR-22	When user changes learning mode, the system will revert to the first question
BR-23	Blocked learner, question set, and group can still be viewed by the Administrator
BR-24	Deleted learner will still have their public question set visible to other learners
BR-25	Pomodoro timer will be reset if user reloads the page
BR-26	Each report can only be resolved once

BR-27	Guests will not have their question set progress tracked by the system.
BR-28	Question set with no subjects will be considered as "Others"
BR-29	Only practice mode will update learner's question set progress
BR-30	The system only shows learners' 10 most recent quiz attempts.

IV. Software Design Description

1. System Design

1.1 System Architecture

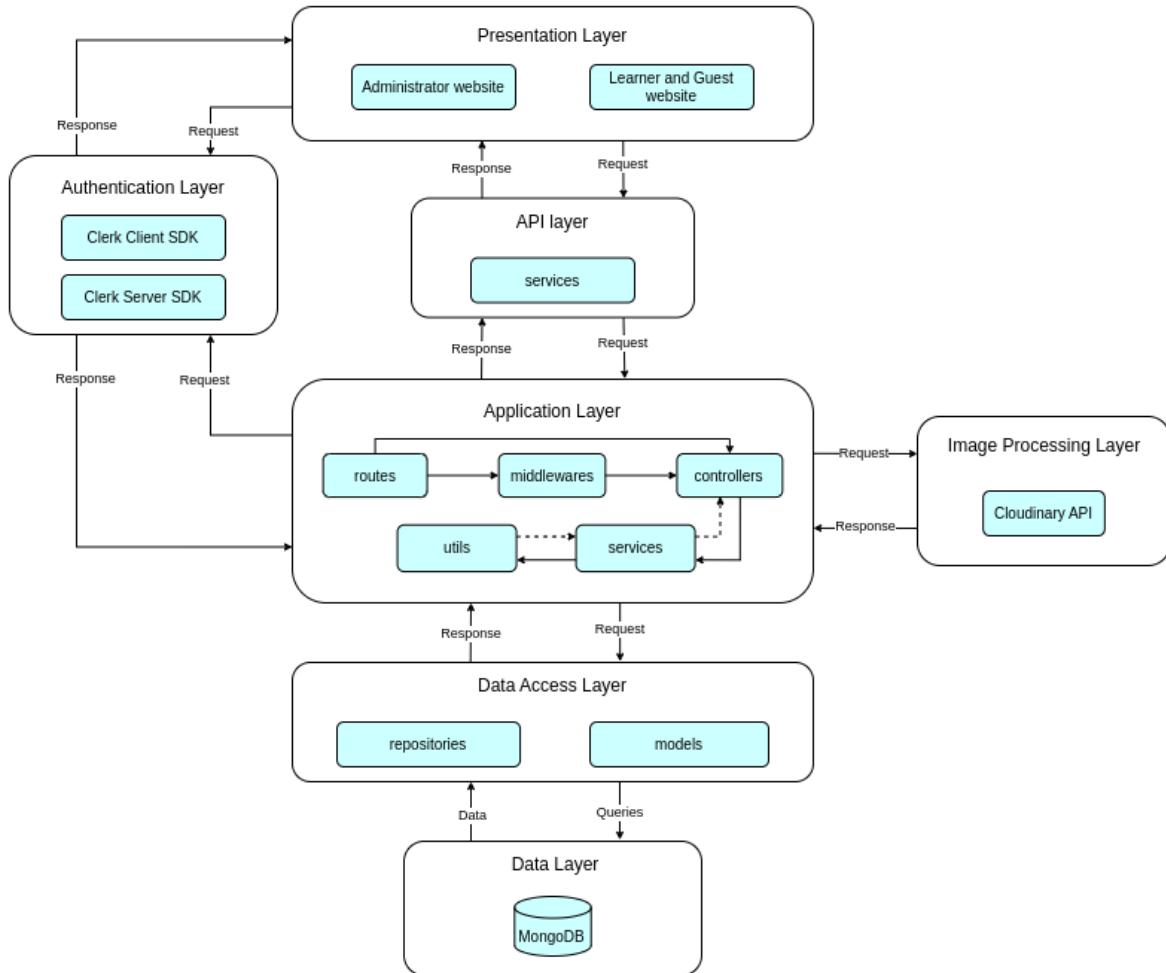


Figure 1.1.1: System architecture³⁰

Presentation Layer (Client-side):

The Presentation Layer serves as the system's interface that interacts with the user and handles user experience through role-specific interfaces. This layer is responsible for rendering visual elements and managing user interactions across different platforms.

- *Administrator Website:*
 - A dedicated web interface providing exclusive access to system administration tools and tasks.
- *Learner and Guest Website:*
 - A shared web interface offering system features and services.
 - Learners have full access, while guests experience restricted functionality.

³⁰ [system-architecture.drawio.png](#)

API Layer (Client-side):

The API layers responsible for handling API calls from the client to the server. It further acts as the bridge between the Presentation layer and the Application layer and abstracts the API interaction of the Presentation layout.

- *services*: contains functions that handle API interaction, managing data fetching and system communication.

Application Layer (Server-side):

The Application layer is where the core business logic of the system is implemented. It receives requests from the API layer, processing them, while interacting with the Data Access Layer.

- *routes*: contain API endpoints and routes requests to the controllers.
- *middlewares*: contain functions that handle request validation, authorization... before sending it to the controllers.
- *controllers*: handle the request / response logic while sending the data to services.
- *services*: contain functions that apply business logic and data processing.
- *utils*: contain helper functions used by services.

Data Access Layer (Server-side):

The Data Access Layer mainly handles the interaction with the database. Acts as a bridge between Application layer and Data layer, helps separate business logic and database-specific operations.

- *repositories*: contain functions that perform database operations using Mongoose pre-defined models.
- *models*: contain schemas created using Mongoose ODM that defines the structure of data stored inside MongoDB.

Data Layer (Server-side):

The Data Layer is where the system manages the actual data storage. It provides data persistence and retrieval capabilities for the system.

- *MongoDB*: The NoSQL database that stores data using flexible, JSON-like documents.

Authentication Layer (External service):

The Authentication Layer handles user authentication and authorization using Clerk, a third-party service.

- *Clerk Client SDK*: integrated into the Presentation layer for client-side authentication and basic authorization.
- *Clerk Server SDK*: integrated into the Application layer for server-side custom authorization.

Image Processing Layer (External service):

The Image Processing Layer handles how images are processed and stored using Cloudinary, a third-party service

- *Cloudinary API*: Used along with the Application layer to handle image processing and storing.

1.2 Package Diagram

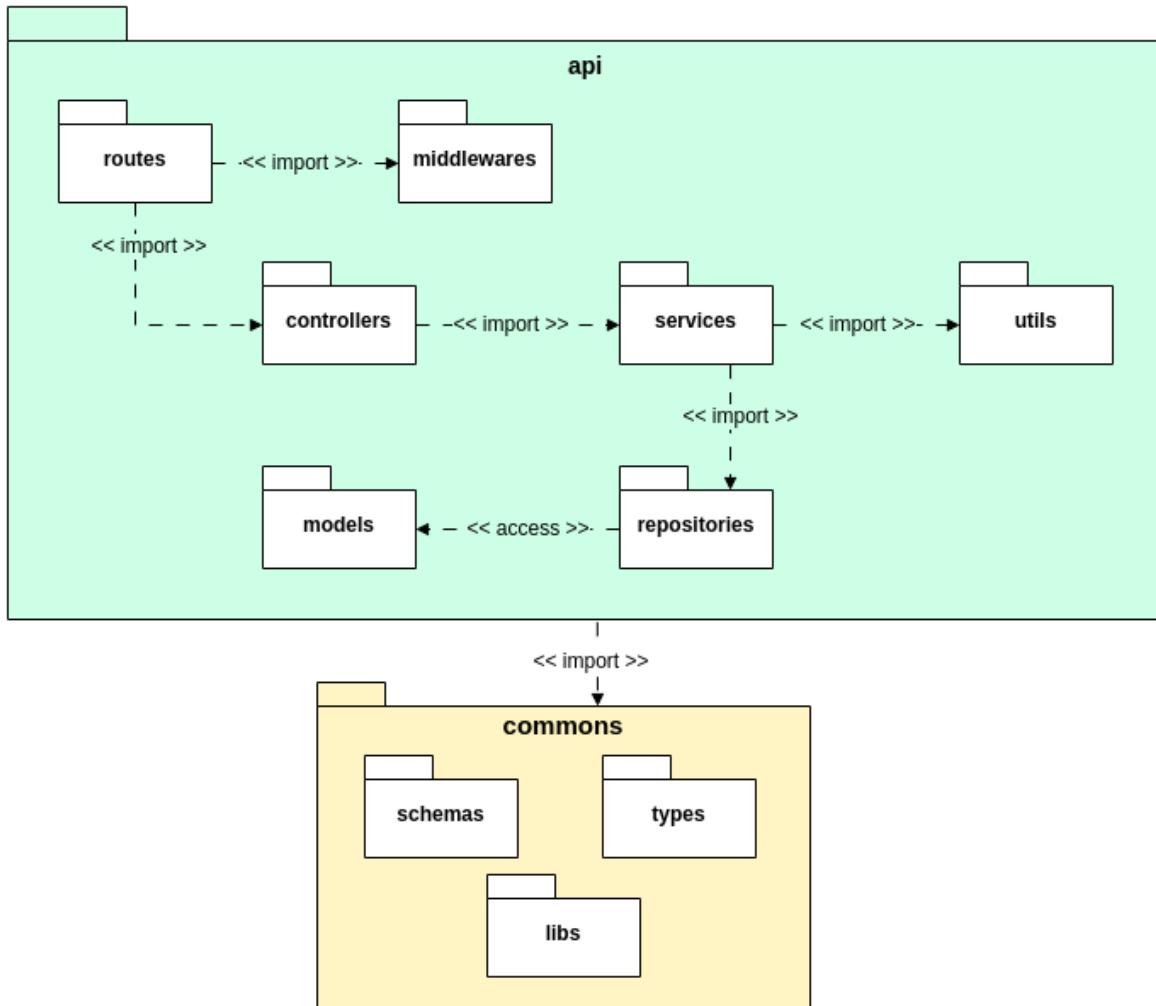


Figure 1.1.2.a: Server-side package diagram³¹

³¹ [server-package-diagram.drawio.png](#)

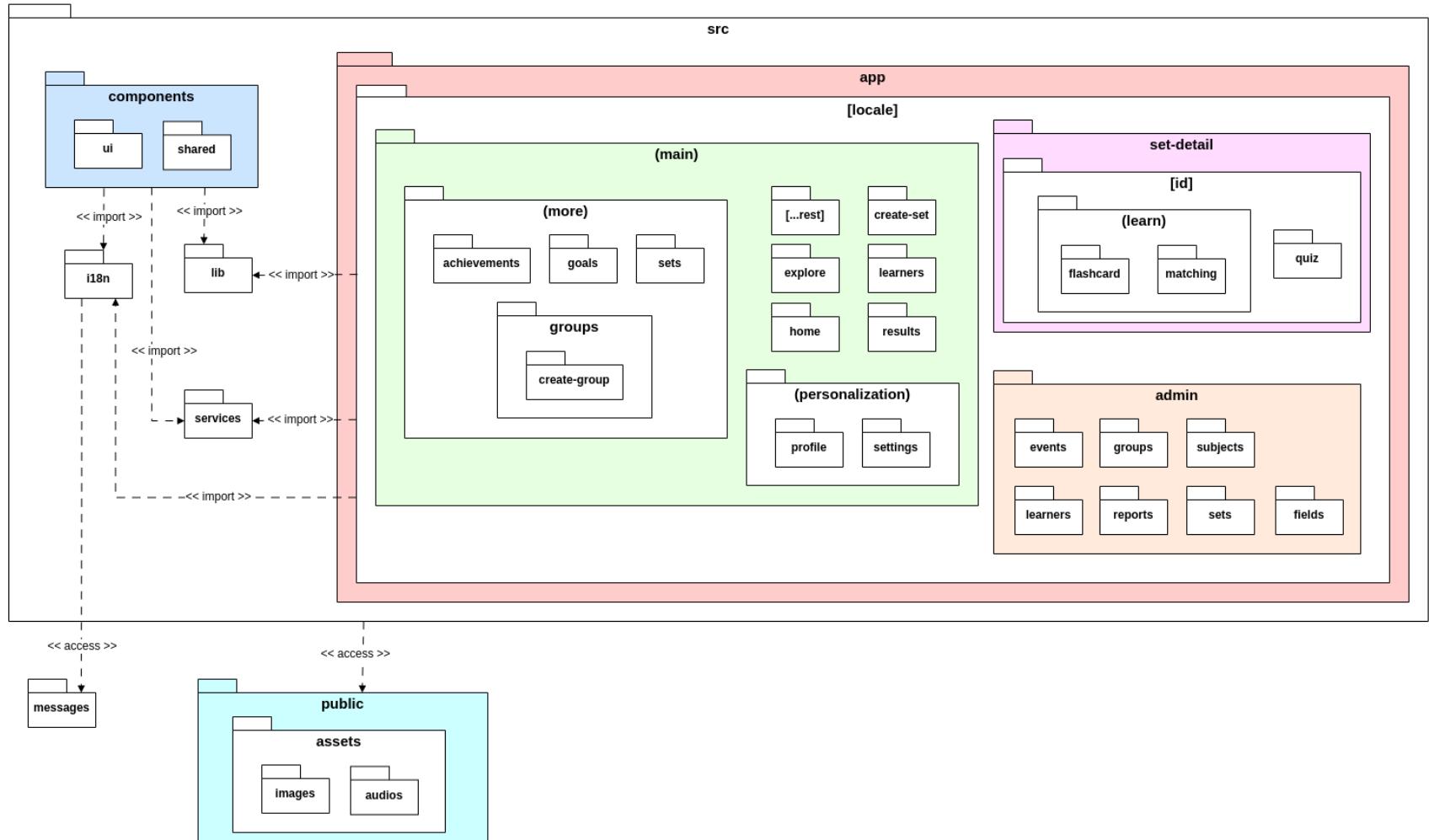


Figure 1.1.2.b: Client-side website package diagram³²

³² [web-package-diagram.drawio.png](#)

No.	Package	Description
01	api	This package handles HTTP requests and responses, while acting as the entry point of the system.
02	commons	This package contains utility sub-package to store data schemas, interfaces and processing data.
03	routes	This package contains lists of API entry points based on the main entities, routing requests through system middlewares and controllers.
04	middlewares	This package contains functions that act as system middleware which validate request data before passing it to the controllers.
05	controllers	This package contains functions that handle request and response while communicating with services.
06	services (server)	This package contains functions that handle system core business rules, communicate with utils for data processing and return data back to the controllers.
07	utils	This package contains functions that process data sent by repositories and return it back to services.
08	repositories	This package contains functions that specifically interact with the database, whose data is constrained by models and schemas.
09	models	This package defines database structure as models based on system functionalities.
10	schemas	This package contains constraints for request data.
11	types	This package defines properties for data handling functions.
12	libs	This package contains classes for Error handling, used by middlewares and data handling functions.
13	src	This package contains the overall project setup for the client website.
14	app	This package contains the routing logic of the client website.
15	components	This package contains shared components that are used across the client website pages

16	public	This package contains public assets like images, sounds,... for the client website
17	messages	This package contains localized languages for Vietnamese and English appearance.
18	[locale]	This package handles the root routing logic for the client website, including handling internationalization.
19	(main)	This package contains the main routes for learner and guest inside the website client.
20	set-detail	This package handles the routing logic of the question set detail page and its mode.
21	admin	This package handles the routing logic for the Administrator website client.
22	ui	This package contains shared components of the ShadCN UI Library.
23	shared	This package contains shared custom components for website clients.
24	i18n	This package handles language switching and routing.
25	services (client)	This package contains integrated functions between the client and the server.
26	lib	This package contains helpers functions for the website client.

2. Database Design

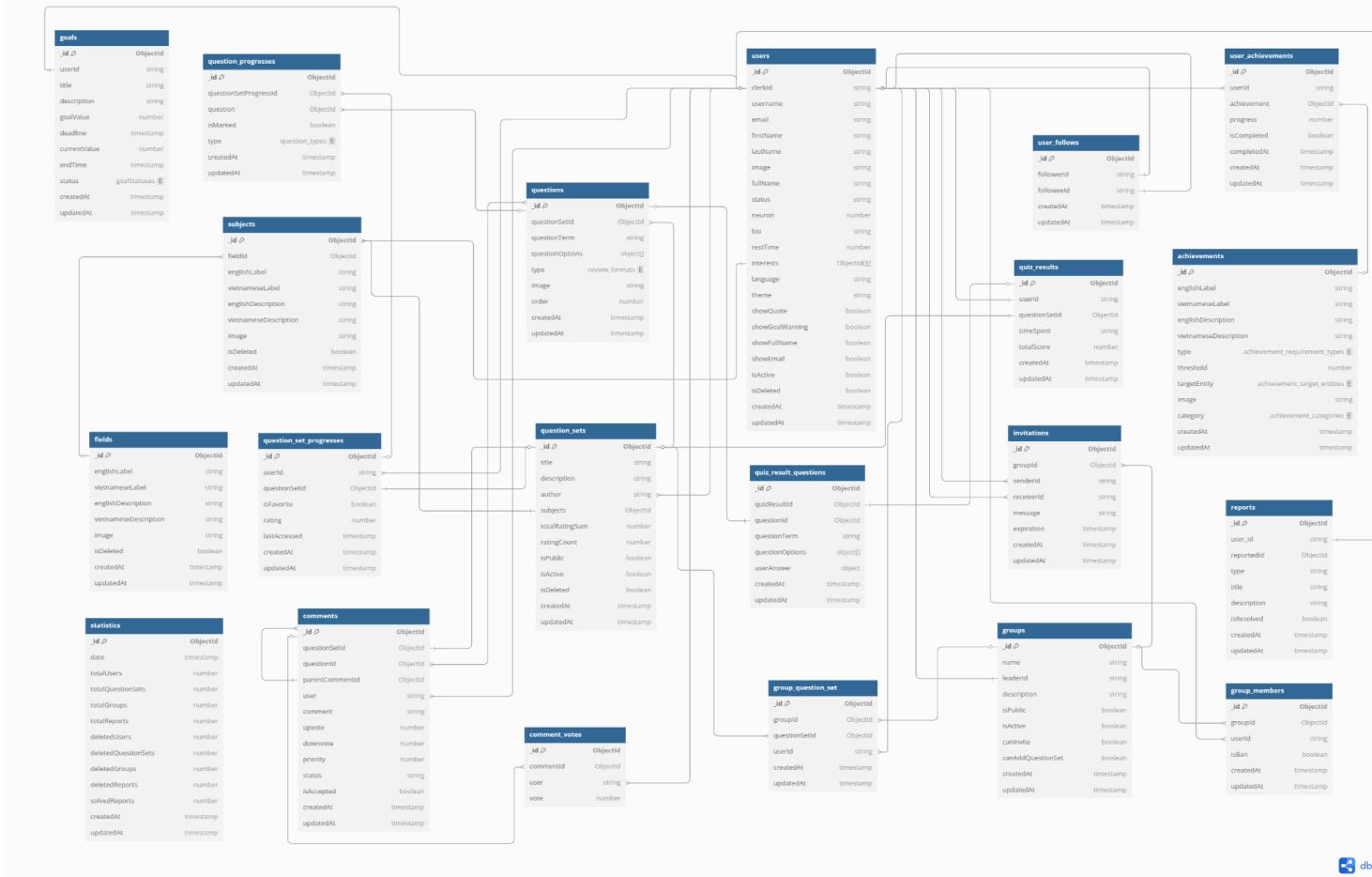


Figure 2. Database design³³

³³ [database-design.png](#)

Database table description

- users table

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
clerkId	VARCHAR	32	X	X		
username	VARCHAR	30	X	X		
email	VARCHAR		X	X		
firstName	VARCHAR	50		X		Default as ""
lastName	VARCHAR	50		X		Default as ""
image	VARCHAR			X		Default as ""
fullName	VARCHAR	100		X		Compute from "firstName" + "lastName"
status	VARCHAR	100		X		Default as ""

neuron	INTEGER			X		Default as 0
bio	VARCHAR	500		X		Default as ""
restTime	INTEGER			X		Default as 0
interests	UUID[]				FK	Reference to subjects table
language	ENUM			X		Default as "en"
theme	ENUM			X		Default as "light"
showQuote	BOOLEAN			X		Default as true
showGoalWarning	BOOLEAN			X		Default as true
showFullName	BOOLEAN			X		Default as true
showEmail	BOOLEAN			X		Default as true
isActive	BOOLEAN			X		Default as true

isDelete	BOOLEAN			X		Default as false
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **user_follows table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
followerId	VARCHAR	32		X	FK	Reference to users table
followeeId	VARCHAR	32		X	FK	Reference to users table
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **user_achievements table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
userId	VARCHAR	32		X	FK	Reference to users table
achievement	UUID			X	FK	Reference to achievements table
progress	INTEGER			X		Default as 0
completed	BOOLEAN			X		Default as false
completeAt	TIMESTAMP					
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- achievements table

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
englishTitle	VARCHAR	100		X		
vietnameseTitle	VARCHAR	100		X		
englishDescription	VARCHAR	500		X		Default as ""
vietnameseDescription	VARCHAR	500		X		Default as ""
type	ENUM			X		Enum achievement_requirement_types { "RECEIVED_MARKING" "COMPLETION" "CREATION" "ACTIVATION" "JOINING" }
threshold	INTEGER			X		

targetEntity	ENUM			X		Enum achievement_target_entities { "QUESTION_SET" "QUIZ" "GROUP" "COMMENT" "BADGE" }
image	VARCHAR			X		
category	ENUM			X		Enum achievement_categories { "LEARNING" "CREATION" "SOCIAL" "MILESTONE" }
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **question_sets table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM

title	VARCHAR	100		X		
description	VARCHAR	500		X		Default as ""
author	VARCHAR	32		X	FK	Reference to users table
subjects	UUID[]				FK	Reference to subjects table
totalRatingSum	INTEGER			X		Default as 0
ratingCount	INTEGER			X		Default as 0
isPublic	BOOLEAN			X		Default as true
isActive	BOOLEAN			X		Default as true
isDeleted	BOOLEAN			X		Default as false
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **questions table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
questionSetId	UUID			X	FK	Reference to question_sets table
questionTerm	VARCHAR	500		X		
questionOptions	JSON[]			X		List of option objects
type	ENUM			X		Enum review_formats { "multiple choice" "essay" "matching" }
image	VARCHAR					
order	INTEGER			X		
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **comments table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
questionSetId	UUID			X	FK	Reference to question_sets table
questionId	UUID				FK	Reference to questions table
parentCommentId	UUID				FK	Self reference
user	VARCHAR	32		X	FK	Reference to users table
comment	VARCHAR	1000		X		
upvote	INTEGER			X		Default as 0
downvote	INTEGER			X		Default as 0
status	VARCHAR			X		Default as active
priority	INTEGER			X		Default as 0

isAccepted	BOOLEAN			X		Default as false
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **comment_votes table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
commentId	UUID			X	FK	Reference to comments table
user	VARCHAR	32		X	FK	Reference to users table
vote	INTEGER			X		Default as 0
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **question_set_progresses table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
userId	VARCHAR	32		X	FK	Reference to users table
questionSetId	UUID			X	FK	Reference to question_sets table
isFavorite	BOOLEAN			X		Default as false
rating	INTEGER			X		Default as 0
lastAccessed	TIMESTAMP			X		Default as current date
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **question_progresses table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
questionSetProgressId	UUID			X	FK	Reference to question_set_progresses table
question	UUID			X	FK	Reference to questions table
isMarked	BOOLEAN			X		Default as false
type	ENUM			X		Enum question_types { "not_learned" "learning" "almost_done" "mastered" }
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **quiz_results table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
userId	VARCHAR	32		X	FK	Reference to users table
questionSetId	UUID			X	FK	Reference to question_sets table
timeSpent	VARCHAR			X		Default as ""
totalScore	INTEGER			X		Default as 0
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- `quiz_result_questions` table

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
<code>_id</code>	UUID		X	X	PK	Auto generated by Mongoose ODM
<code>quizResultId</code>	UUID			X		
<code>questionId</code>	UUID			X		
<code>questionTerm</code>	VARCHAR	250		X		
<code>questionOptions</code>	JSON			X		Contains list of options object
<code>userAnswer</code>	JSON			X		Contains user answer object
<code>createdAt</code>	TIMESTAMP			X		Auto generated by Mongoose ODM
<code>updatedAt</code>	TIMESTAMP			X		Auto updated by Mongoose ODM

- goals table

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
title	VARCHAR	100		X		
description	VARCHAR	500				
goalValue	INTEGER			X		Default as 0
deadline	TIMESTAMP					
userId	VARCHAR	32		X	FK	Reference to users table
currentValue	INTEGER			X		Default as 0
endTime	TIMESTAMP			X		
status	ENUM			X		Enum goalStatuses{ Not_start Doing Cancel Finished}

						Overdue }
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- reports table

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
userId	VARCHAR			X	FK	Reference to users table
reportedId	UUID			X	FK	Reference to users table
type	ENUM			X		Enum report_types { "user", "comment", "questionSet" }
title	VARCHAR	100		X		

description	VARCHAR	500				
isResolved	BOOLEAN			X		Default as false
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **groups table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
name	VARCHAR	100		X		
leaderId	VARCHAR	32		X	FK	Reference to users table
description	VARCHAR	750		X		
isPublic	BOOLEAN			X		Default as true

isActive	BOOLEAN			X		Default as true
canInvite	BOOLEAN			X		Default as true
canAddQuestionSet	BOOLEAN			X		Default as true
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **group_members table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
groupId	UUID			X	FK	Reference to groups table
userId	VARCHAR	32		X	FK	Reference to users table
isBan	BOOLEAN			X		Default as false

createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **group_question_sets table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
groupId	UUID			X	FK	Reference to groups table
questionSetId	UUID			X	FK	Reference to question_sets table
userId	VARCHAR	32		X	FK	Reference to users table
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **invitations table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
groupId	UUID			X	FK	Reference to groups table
senderId	VARCHAR	32		X	FK	Reference to group_members table
receiverId	VARCHAR	32		X	FK	Reference to users table
message	VARCHAR	250		X		
expiration	TIMESTAMP			X		Default as 3 days
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- statistics table

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
date	TIMESTAMP			X		
totalUsers	INTEGER			X		
totalQuestionSets	INTEGER			X		
totalGroups	INTEGER			X		
totalReports	INTEGER			X		
deletedUsers	INTEGER			X		
deletedQuestionSets	INTEGER			X		
deletedGroups	INTEGER			X		
deletedReports	INTEGER			X		

solvedReports	INTEGER			X		
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **fields table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
englishLabel	VARCHAR	100	X	X		
vietnameseLabel	VARCHAR	100	X	X		
englishDescription	VARCHAR	500		X		Default as ""
vietnameseDescription	VARCHAR	500		X		Default as ""
image	VARCHAR					

isDeleted	BOOLEAN			X		Default as false
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

- **subjects table**

Attribute	Data type	Size	Unique	Not null	PK/FK	Note
_id	UUID		X	X	PK	Auto generated by Mongoose ODM
fieldId	UUID			X	FK	Reference to fields table
englishLabel	VARCHAR	100	X	X		
vietnameseLabel	VARCHAR	100	X	X		
englishDescription	VARCHAR	500		X		Default as ""
vietnamesedescription	VARCHAR	500		X		Default as ""

image	VARCHAR					
isDeleted	BOOLEAN			X		Default as false
createdAt	TIMESTAMP			X		Auto generated by Mongoose ODM
updatedAt	TIMESTAMP			X		Auto updated by Mongoose ODM

3. Detailed Design

3.1 User Authentication

3.1.1 Sign in

3.1.1.1 Class diagram

3.1.1.1.1 Sign in with credential

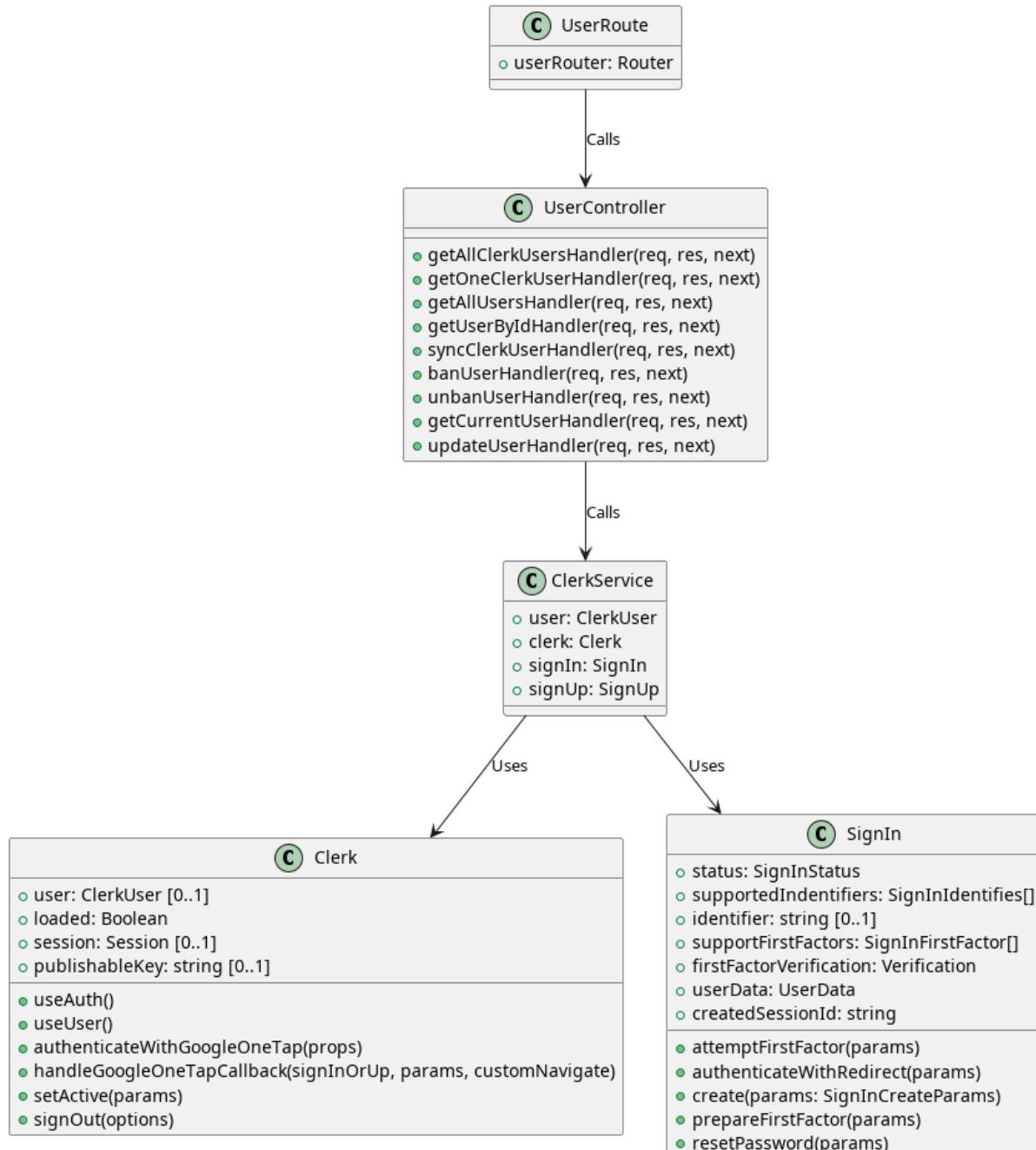


Figure 3.1.1.1.1: Sign in with credentials³⁴

³⁴ [sign in with credentials.png](#)

3.1.1.1.2 Sign in with OTP

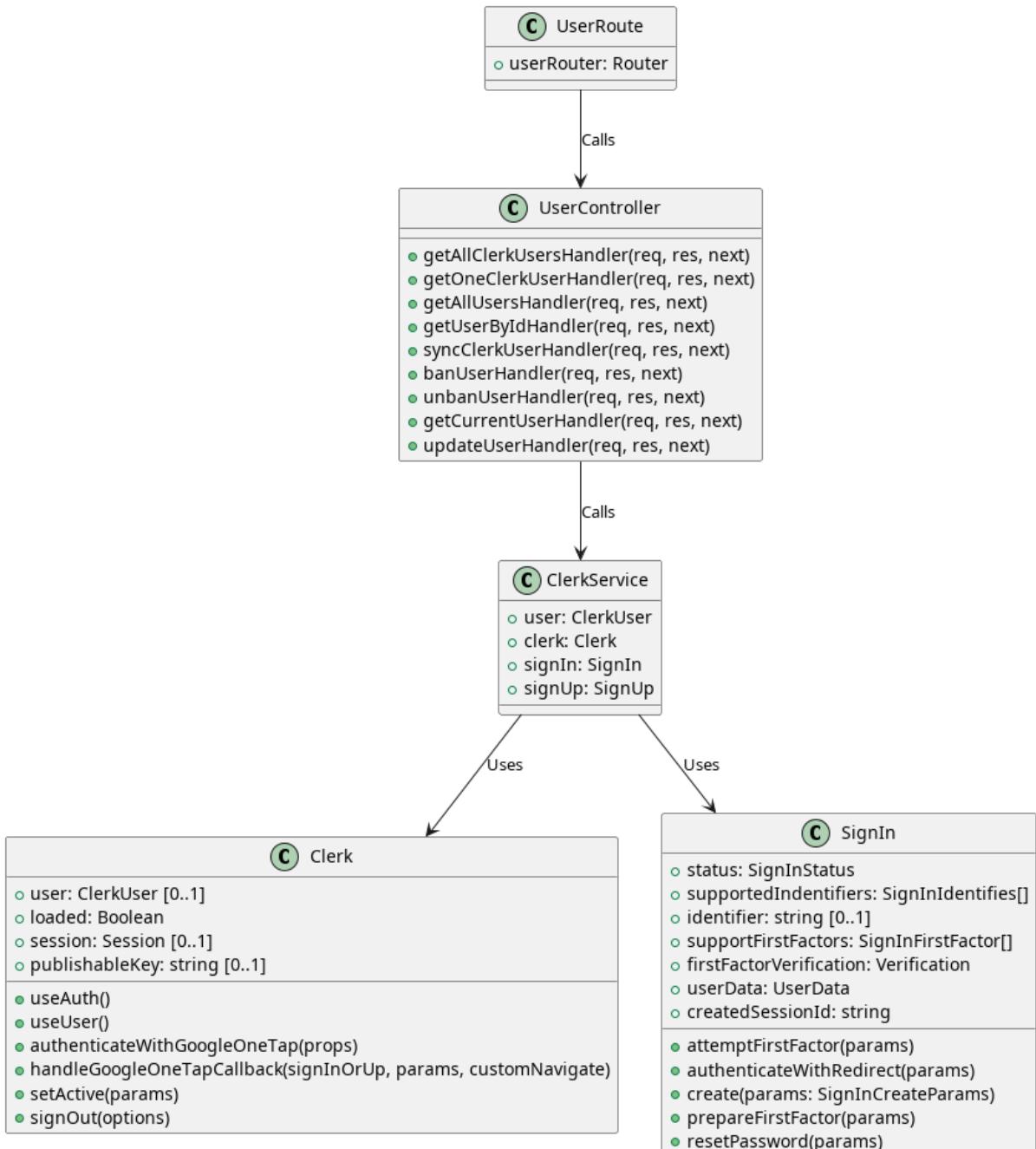


Figure 3.1.1.1.2: Sign in with OTP³⁵

³⁵ [sign in with credentials.png](#)

3.1.1.1.3 Sign in with Google

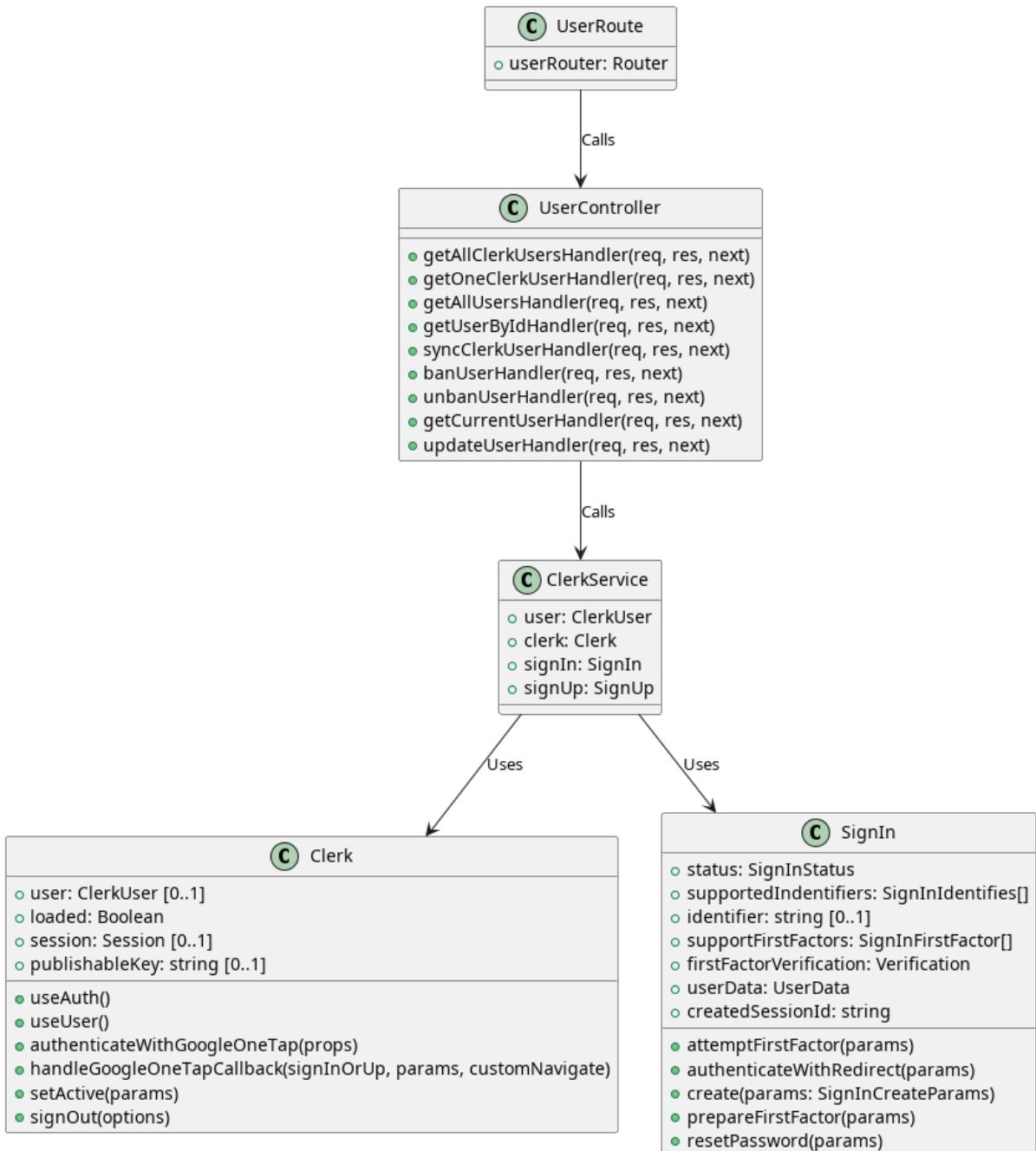


Figure 3.1.1.3: Sign in with Google³⁶

³⁶ [sign in with credentials.png](#)

3.1.1.2 Sequence diagram

3.1.1.2.1 Sign in with credential

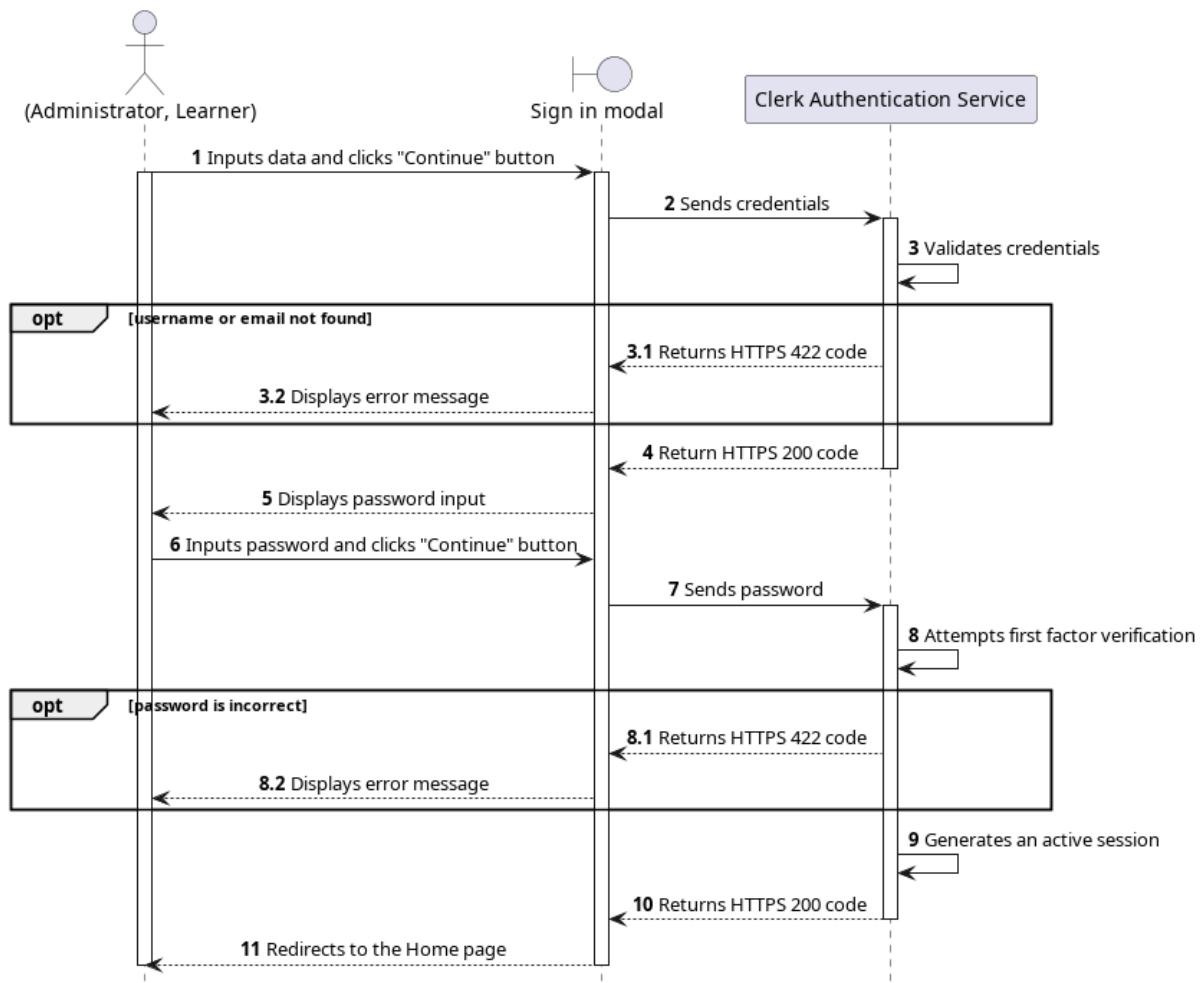


Figure 3.1.1.2.1: Sign in with credentials³⁷

³⁷ [sign in with credentials.png](#)

3.1.1.2.2 Sign in with OTP

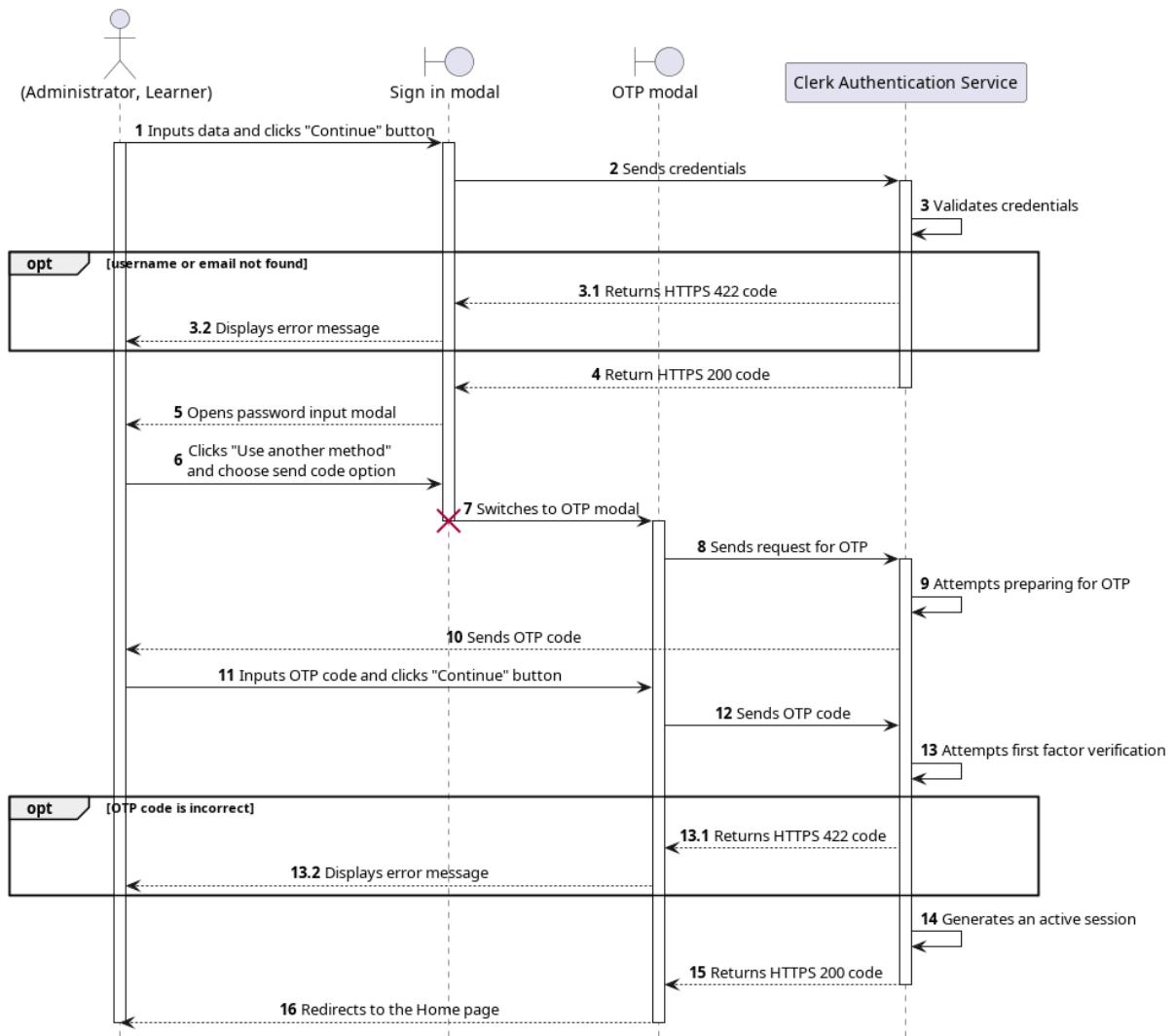


Figure 3.1.1.2.2: Sign in with OTP³⁸

³⁸ [sign in with credential and OTP.png](#)

3.1.1.2.3 Sign in with Google

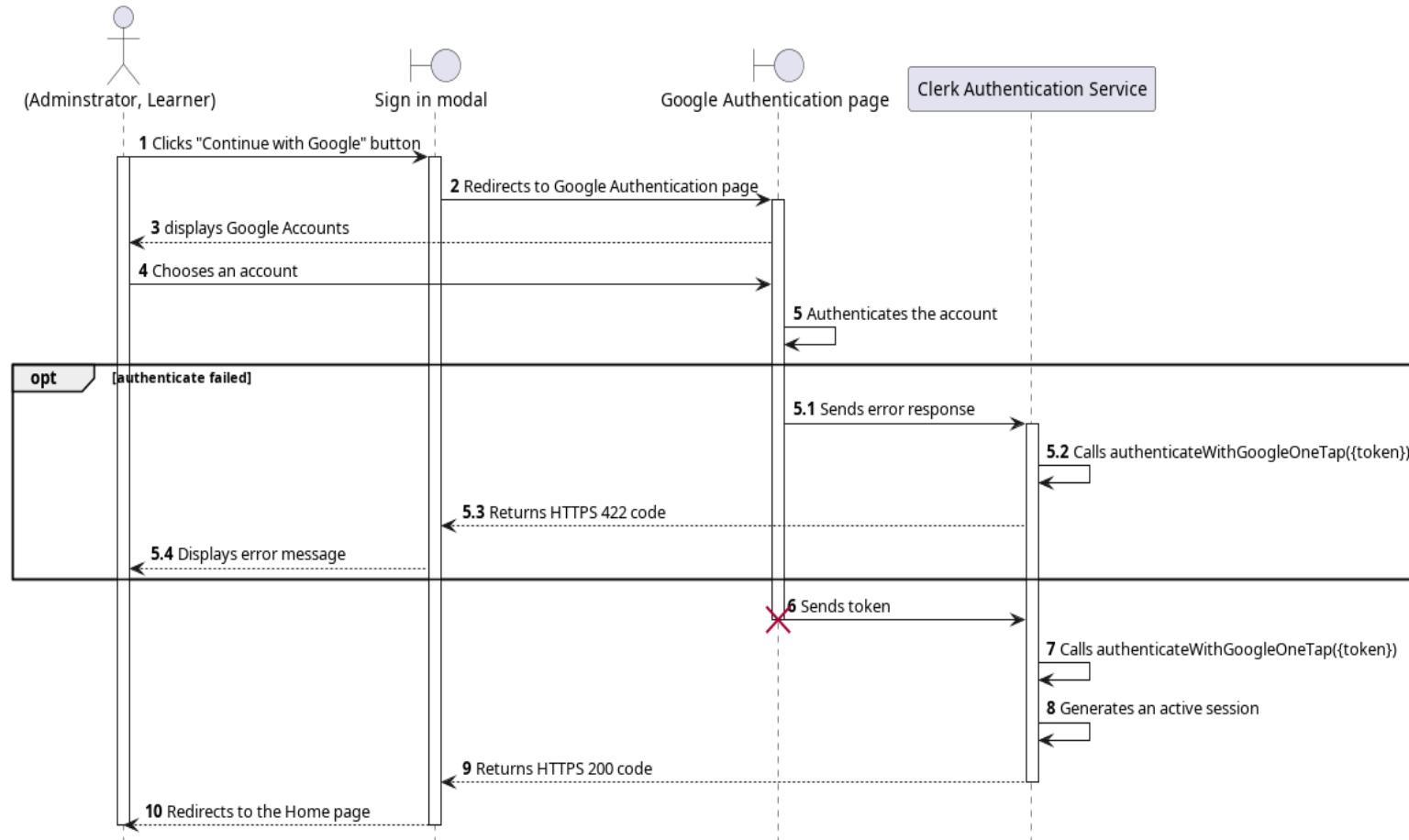


Figure 3.1.1.2.3: Sign in with Google³⁹

³⁹ [sign in with Google.png](#)

3.1.2 Sign up

3.1.2.1 Class diagram

3.1.2.1.1 Sign up with credentials

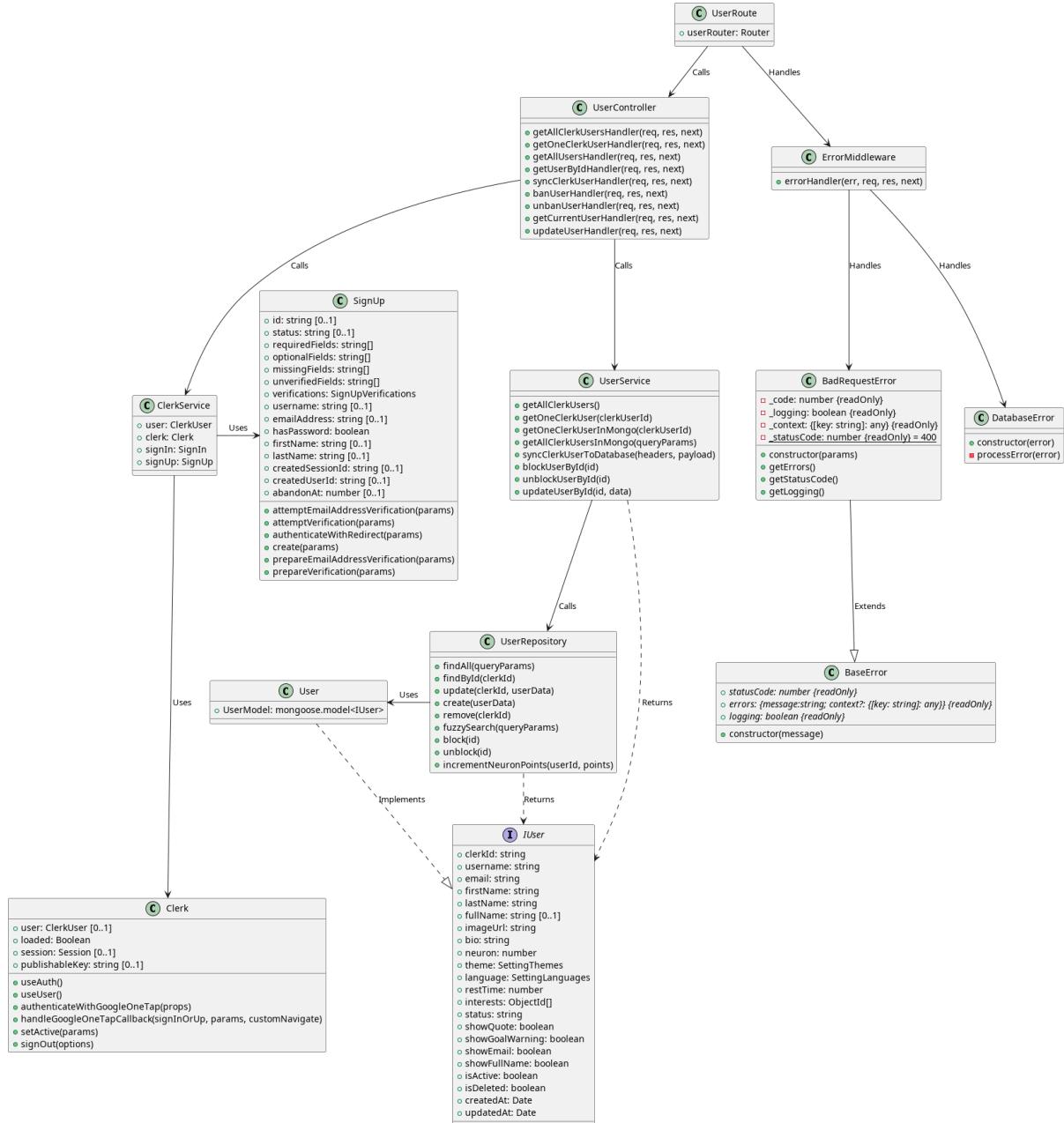


Figure 3.1.2.1.1: Sign up with credentials⁴⁰

⁴⁰ [sign up with credentials.png](#)

3.1.2.1.2 Sign up with Google

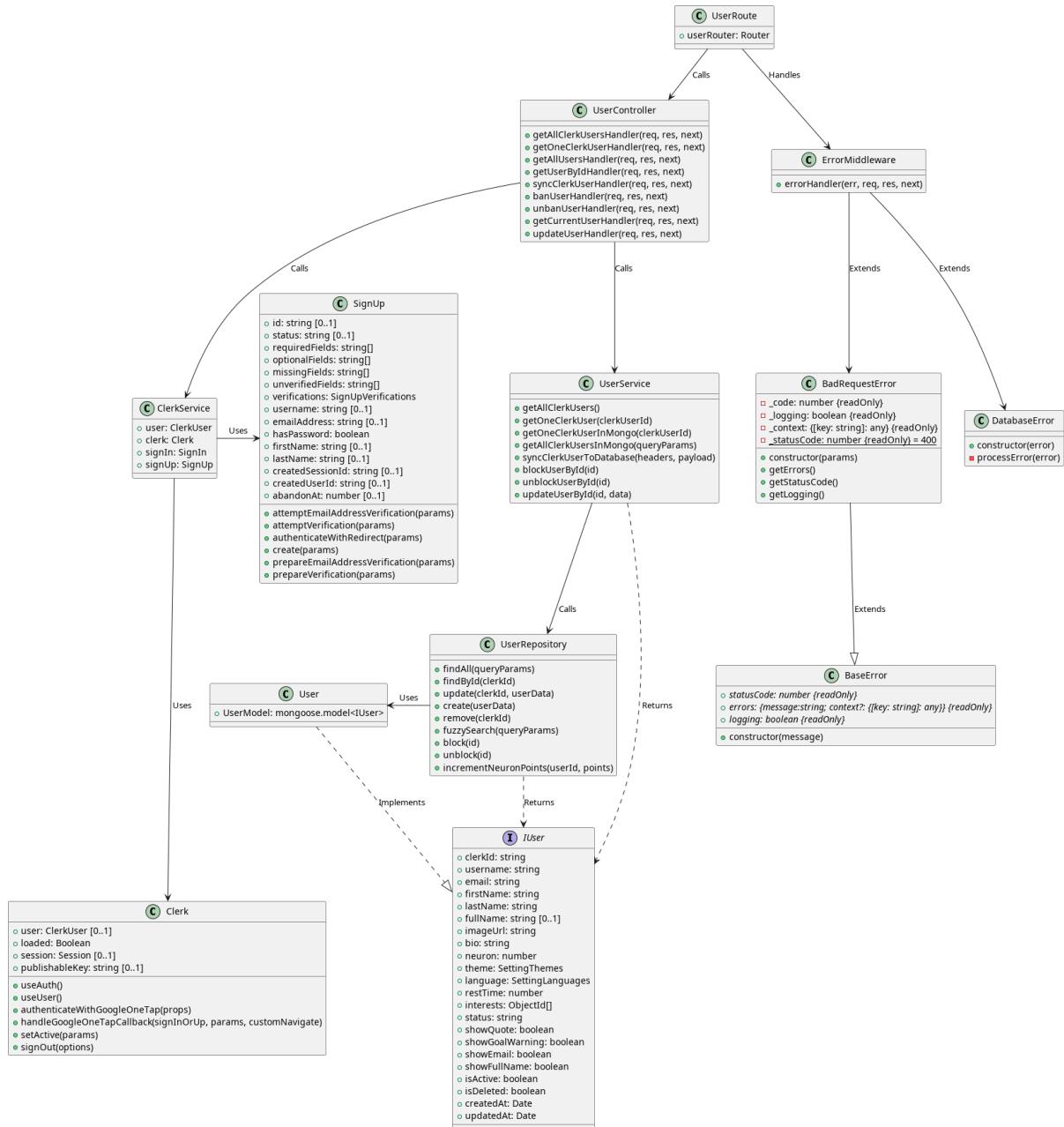


Figure 3.1.2.1.2: Sign up with Google⁴¹

⁴¹ [sign up with Google.png](#)

3.1.2.2 Sequence diagram

3.1.2.2.1 Sign up with credential

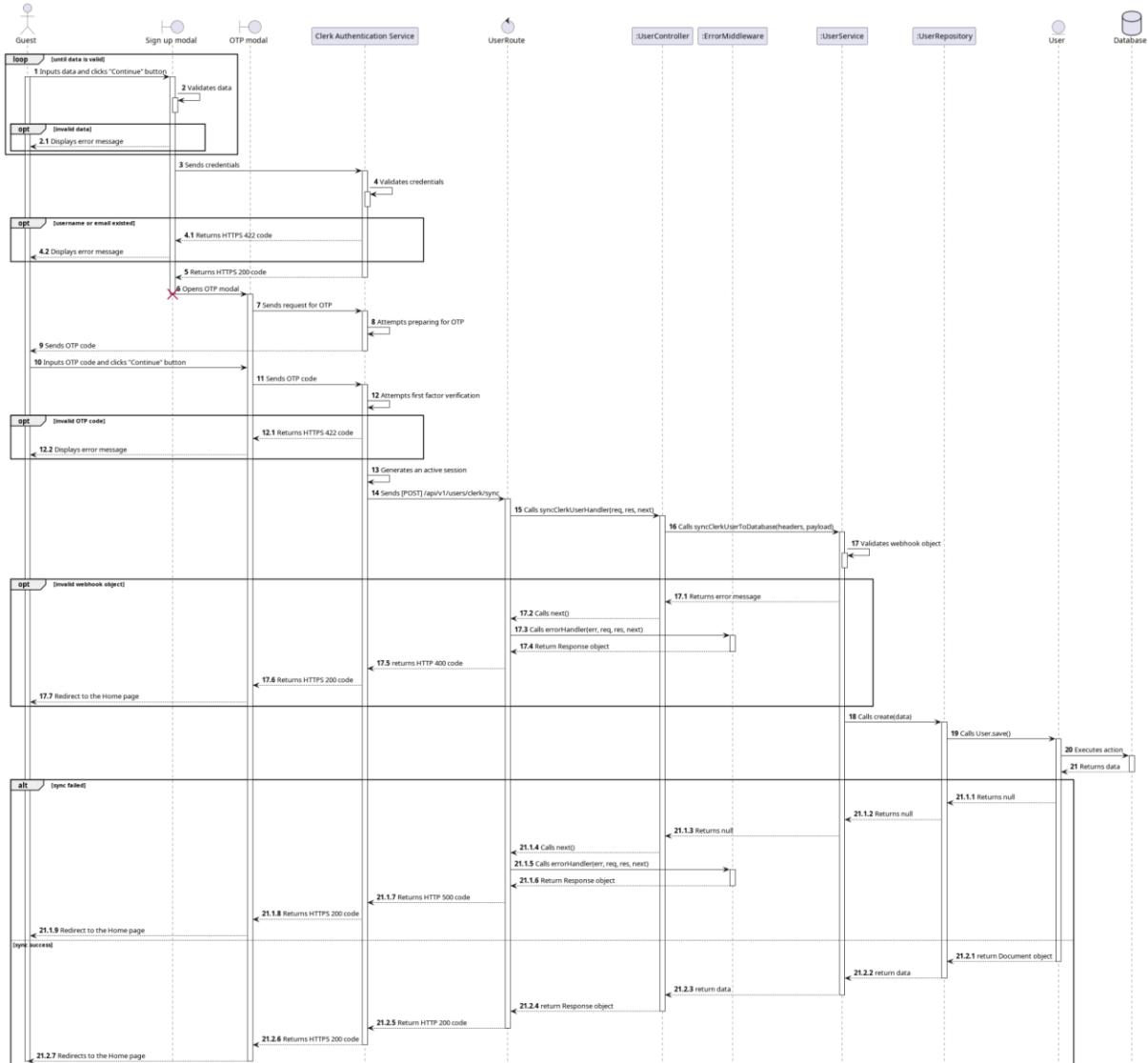


Figure 3.1.2.2.1: Sign up with credentials⁴²

⁴² [sign up.png](#)

3.1.2.2.2 Sign up with Google

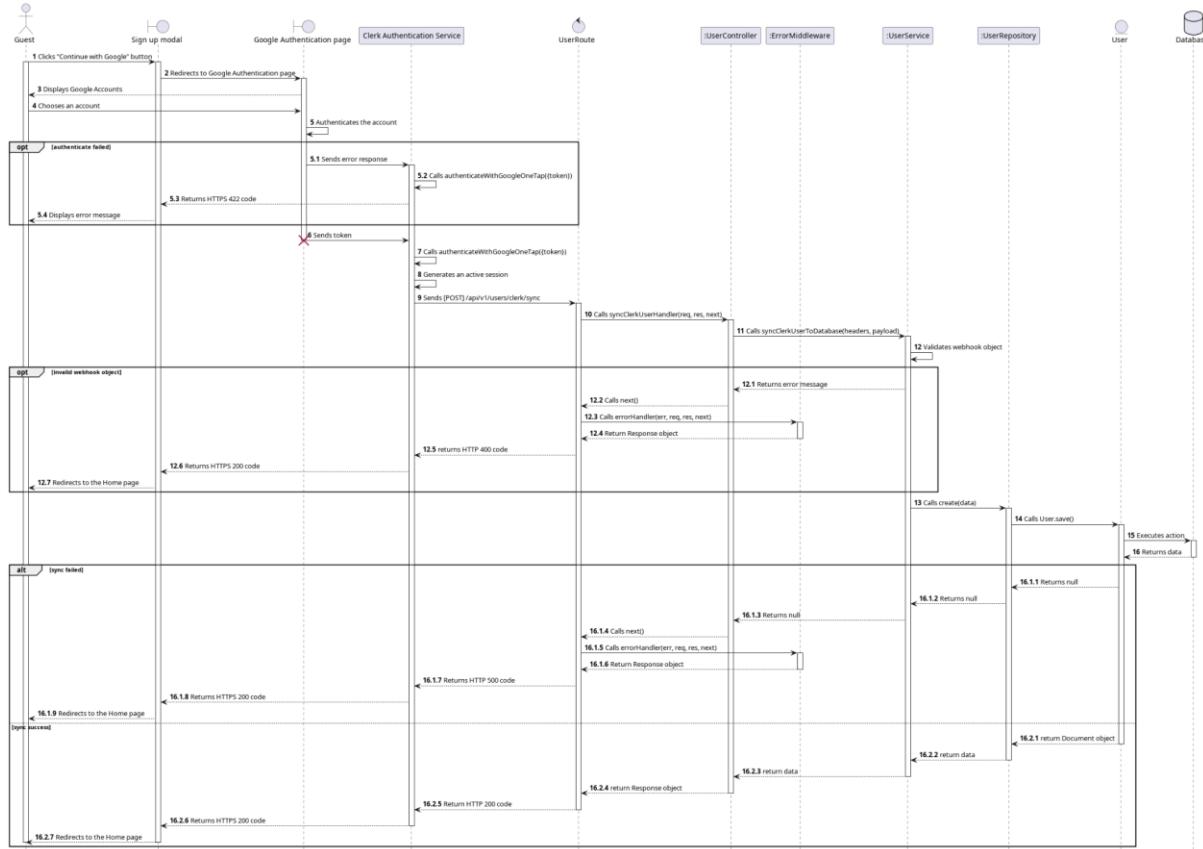


Figure 3.1.2.2.2: Sign up with Google⁴³

⁴³ sign up with Google.png

3.2 Profile Management

3.2.1 View account profile

3.2.1.1 Class diagram

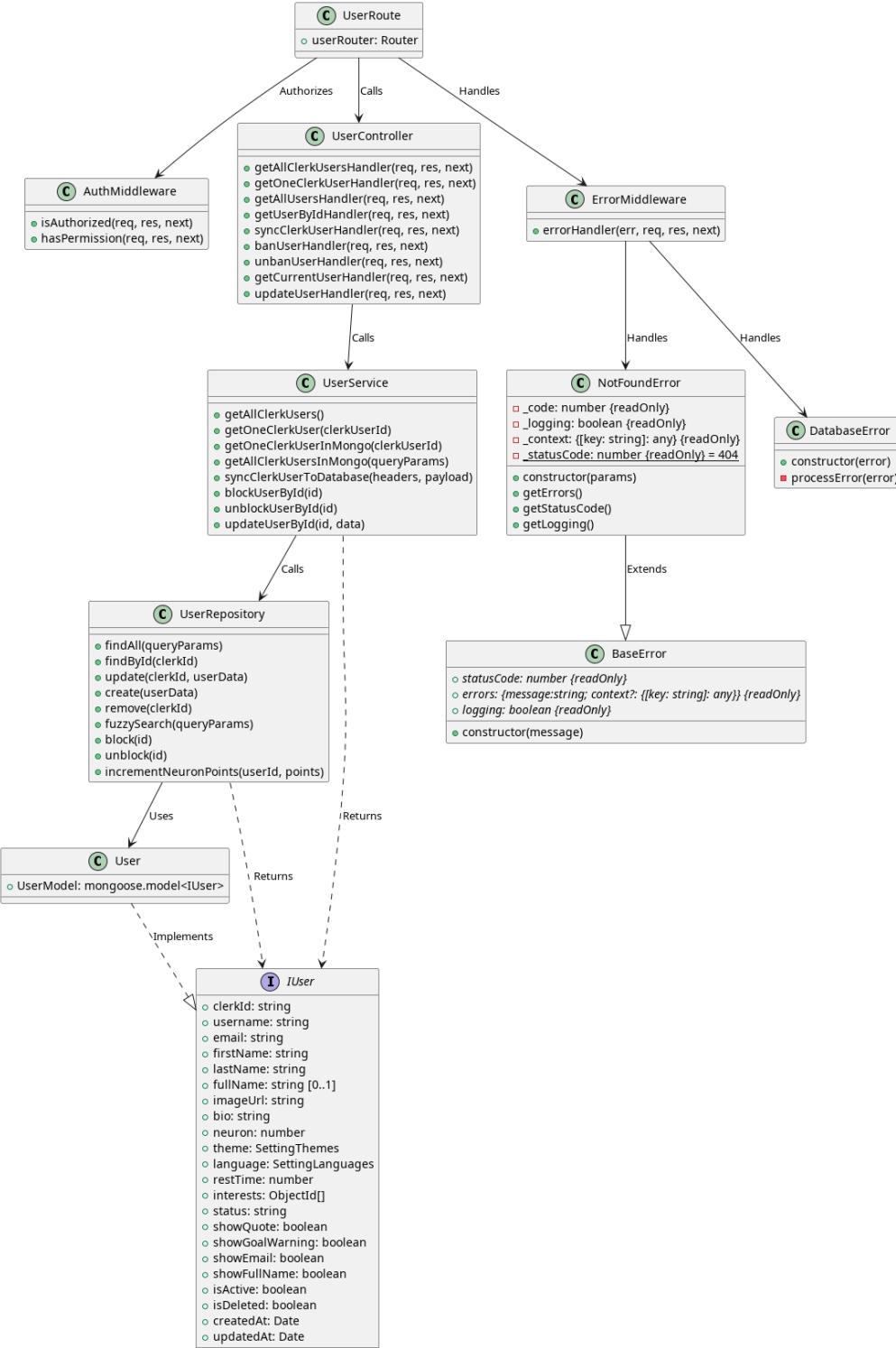


Figure 3.2.1.1.a: Learner view account profile⁴⁴

⁴⁴ [learner view profile.png](#)

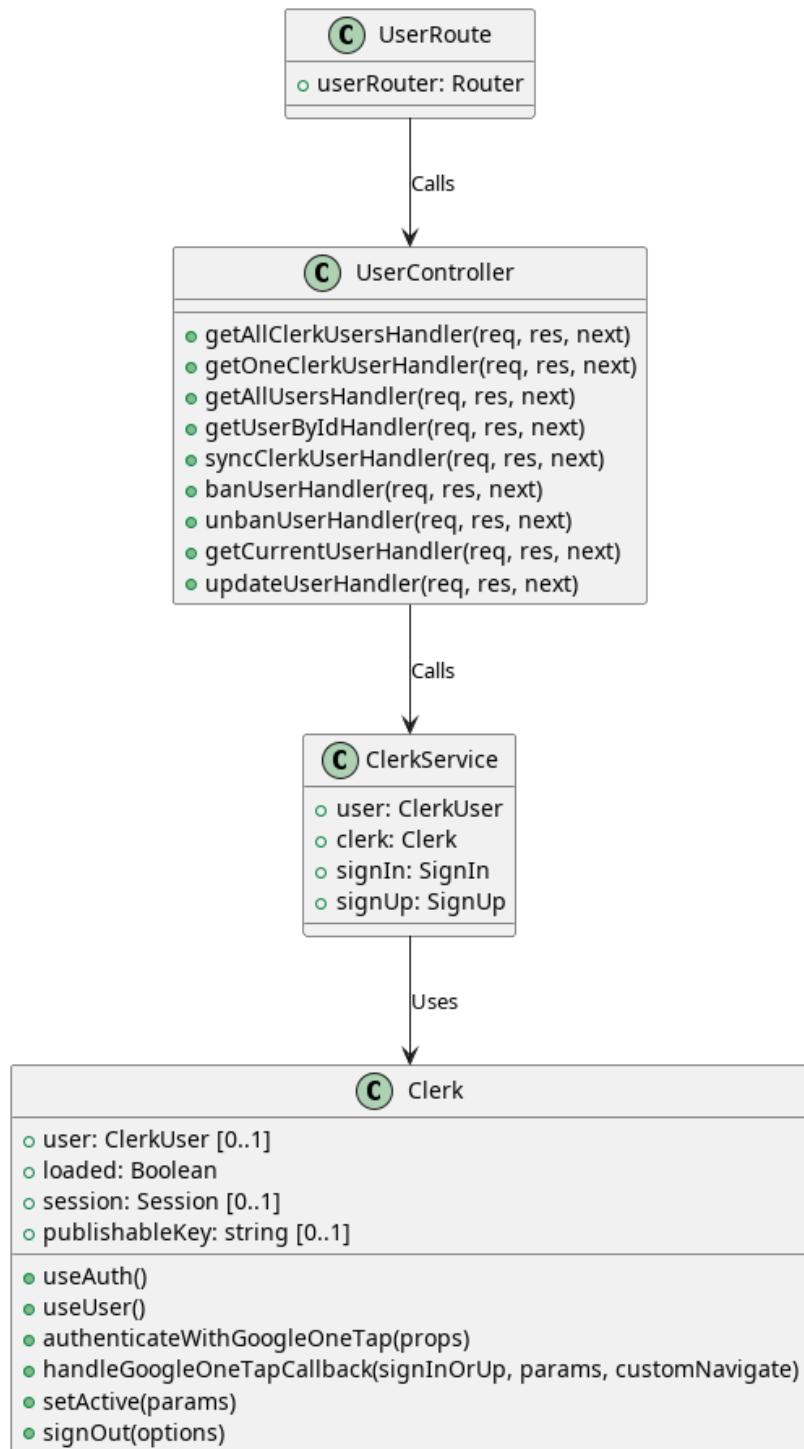


Figure 3.2.1.1.b: Administrator view account profile⁴⁵

⁴⁵ [admin view profile.png](#)

3.2.1.2 Sequence diagram

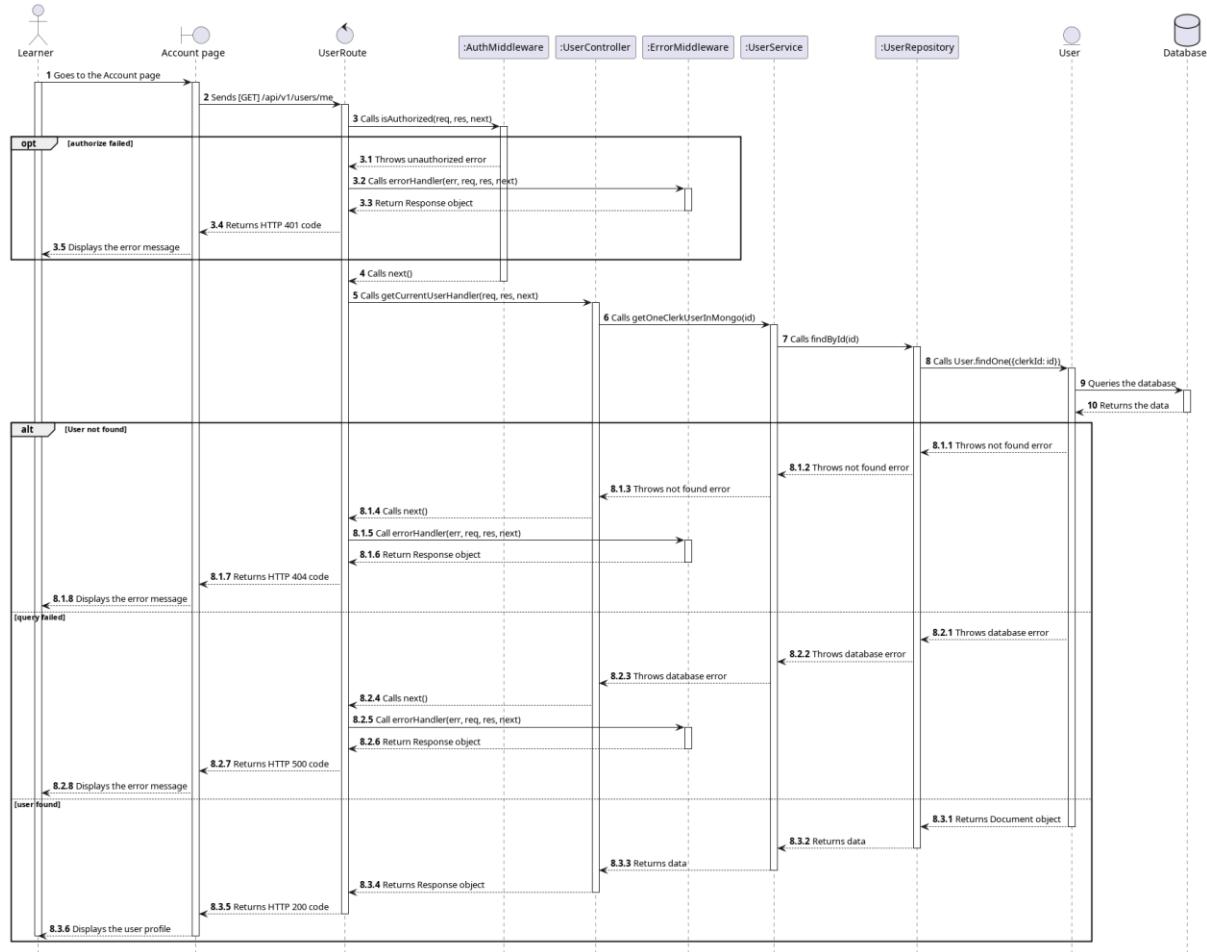


Figure 3.2.1.2.a: Learner view account profile⁴⁶

⁴⁶ [view learner profile.png](#)

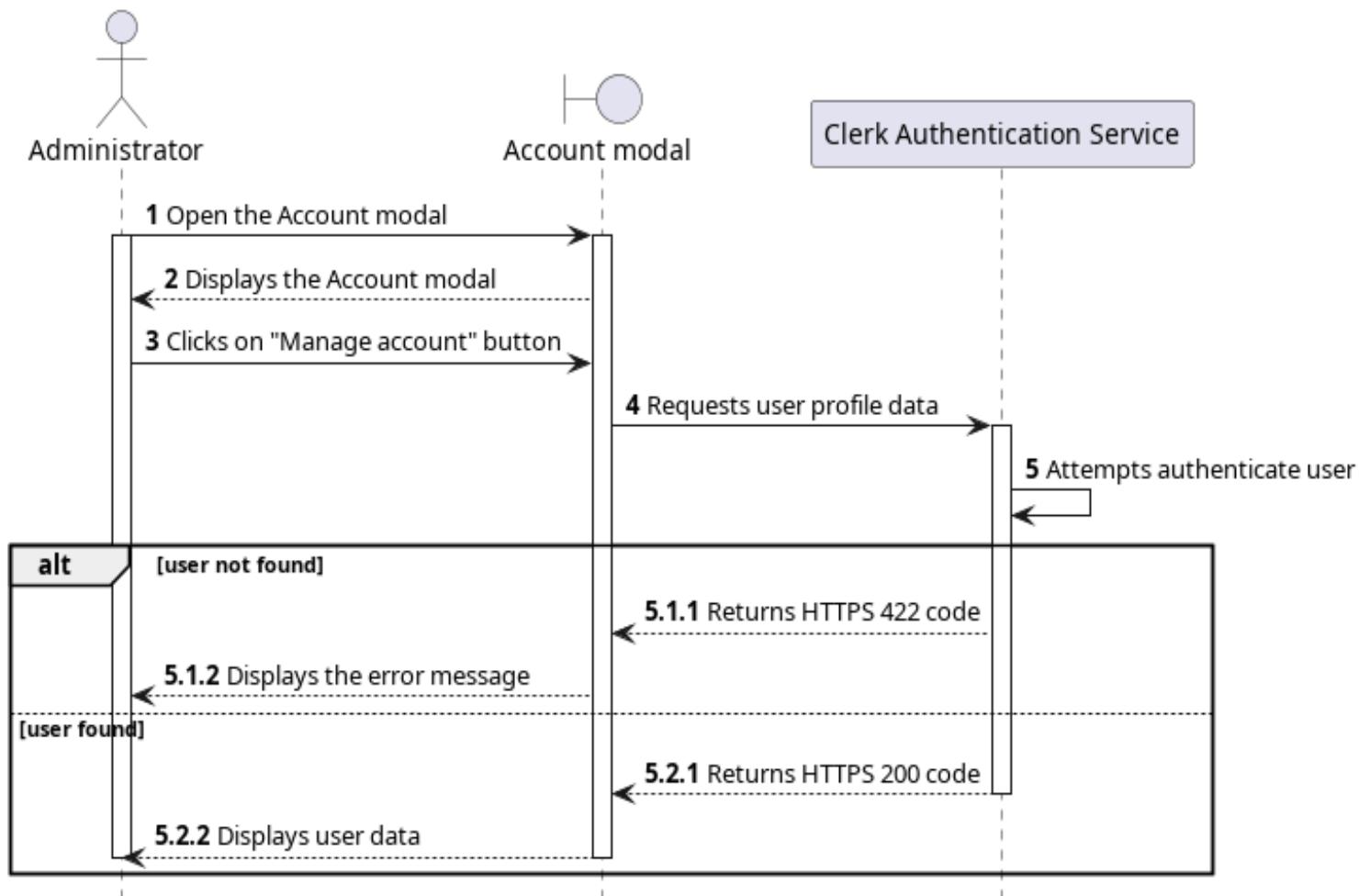


Figure 3.2.1.2.b: Administrator view account profile⁴⁷

⁴⁷ [admin view profile account.png](#)

3.2.2 Update account information

3.2.2.1 Class diagram

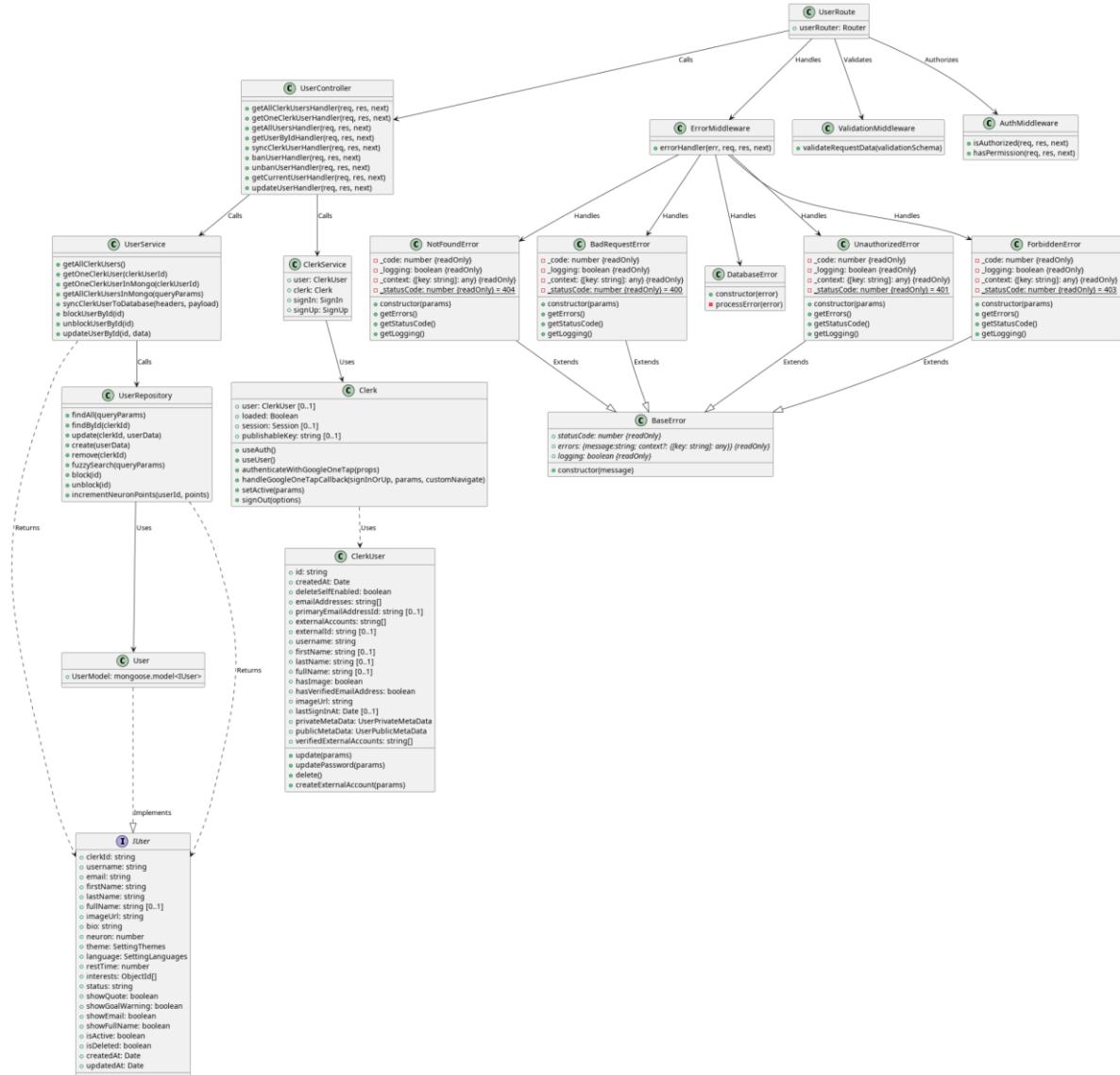


Figure 3.2.2.1: Update account information⁴⁸

⁴⁸ [update profile.png](#)

3.2.2.2 Sequence diagram

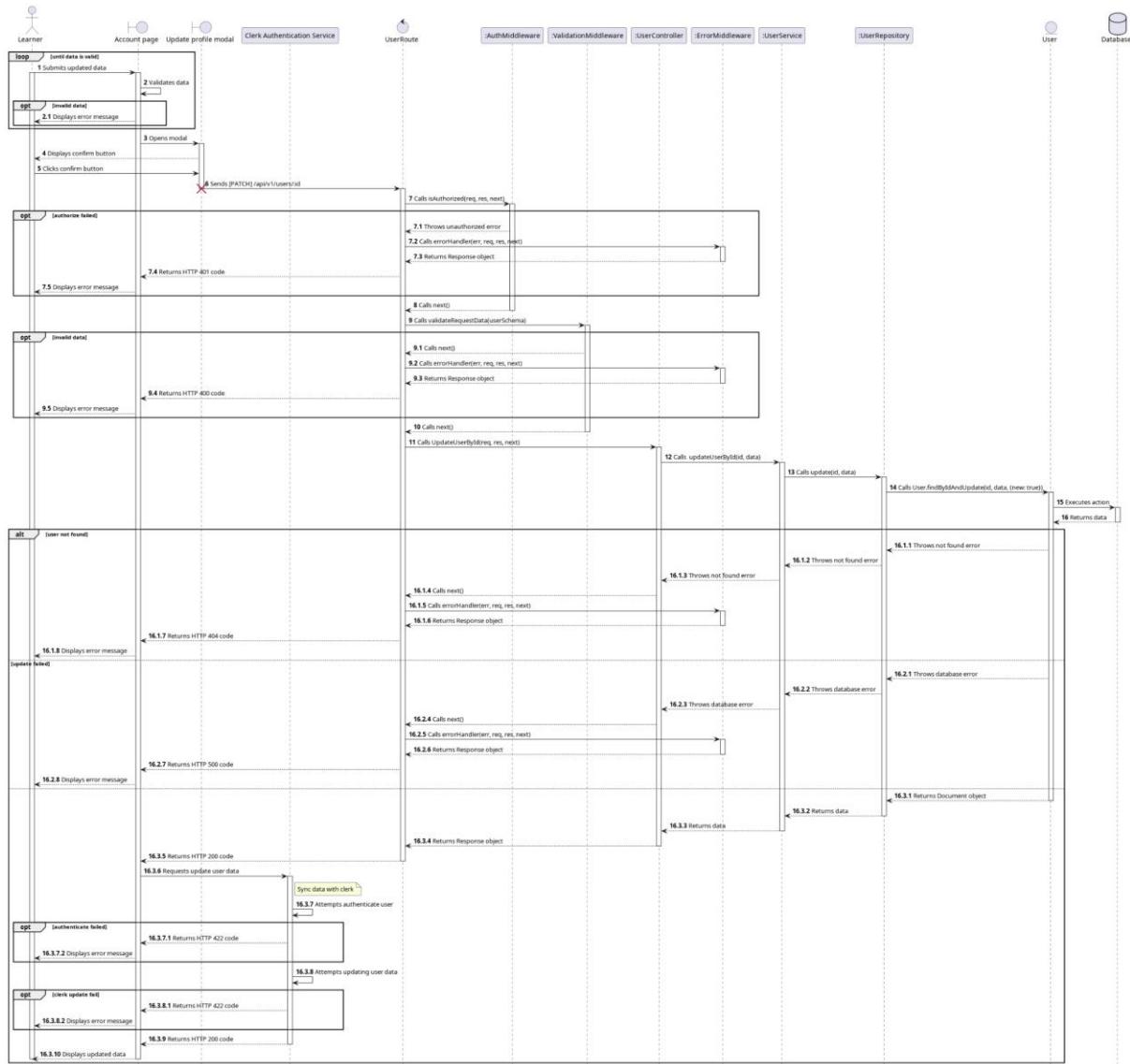


Figure 3.2.2.2.a: Learner update account information⁴⁹

⁴⁹ [learner update profile data.png](#)

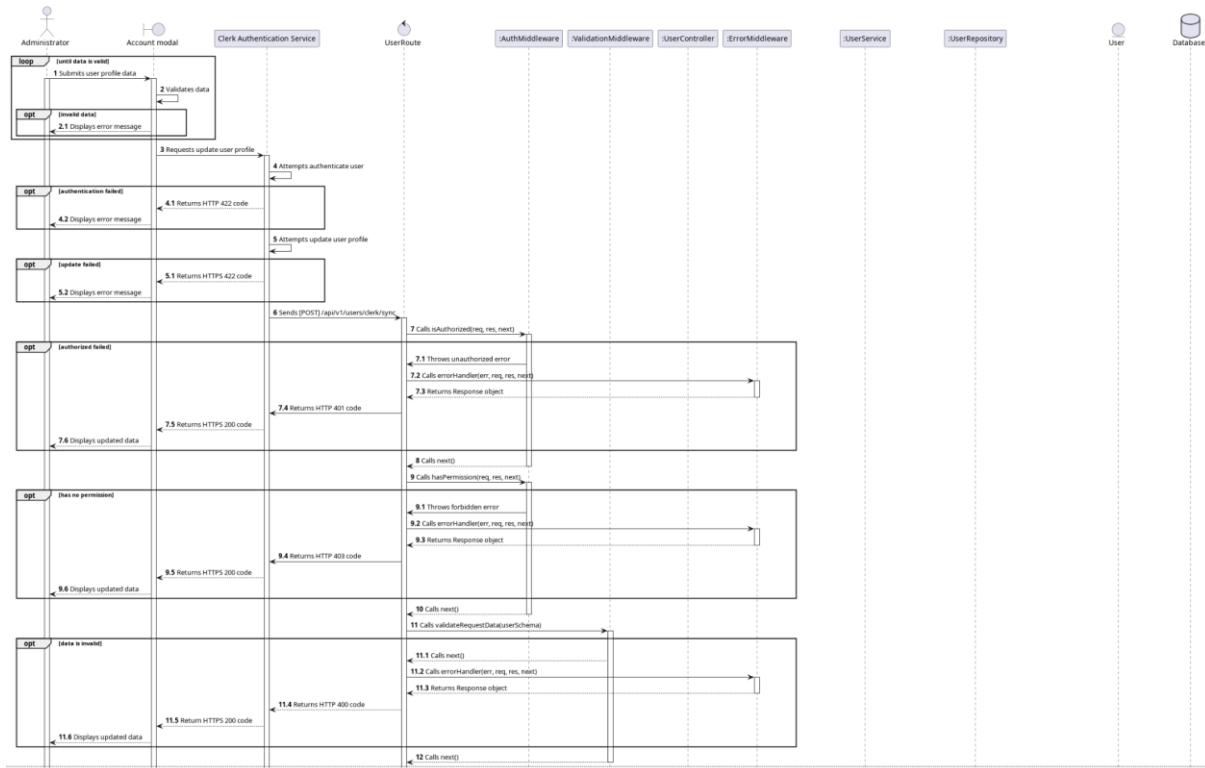


Figure 3.2.2.2.b: Administrator update account information - 1⁵⁰

⁵⁰ [admin update profile.png](#)

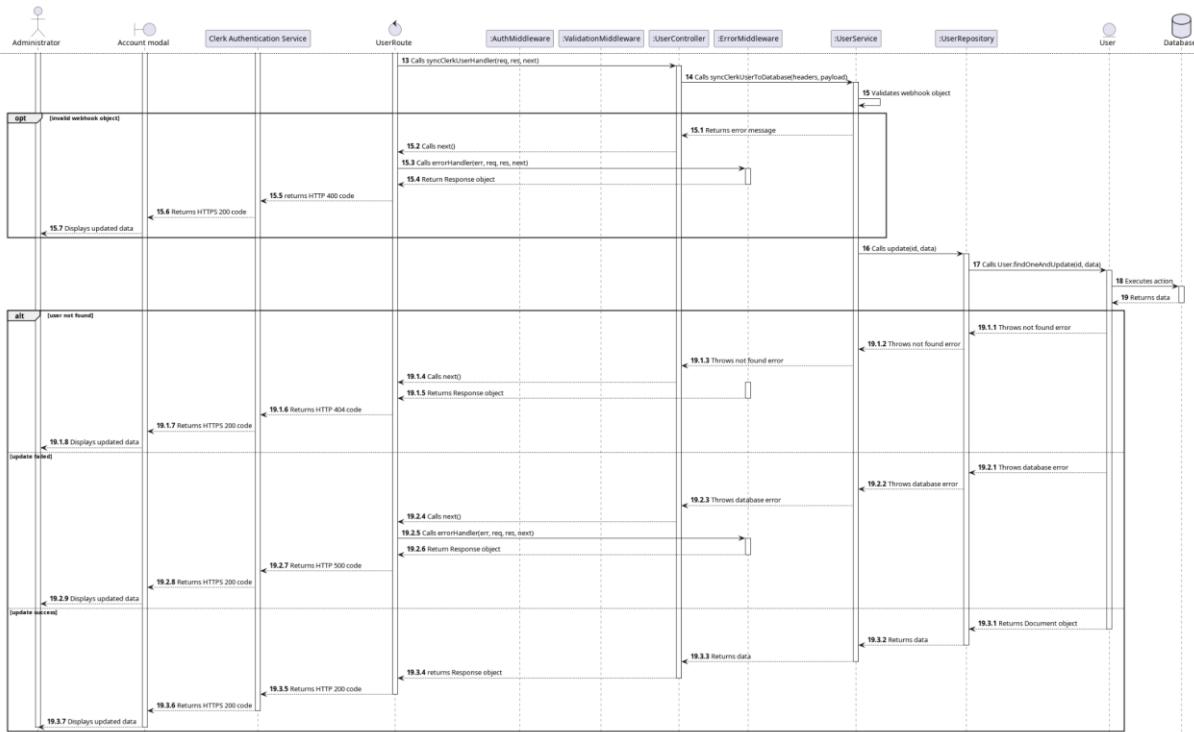


Figure 3.2.2.2.c: Administrator update account information - 2

3.2.3 Update password

3.2.3.1 Class diagram

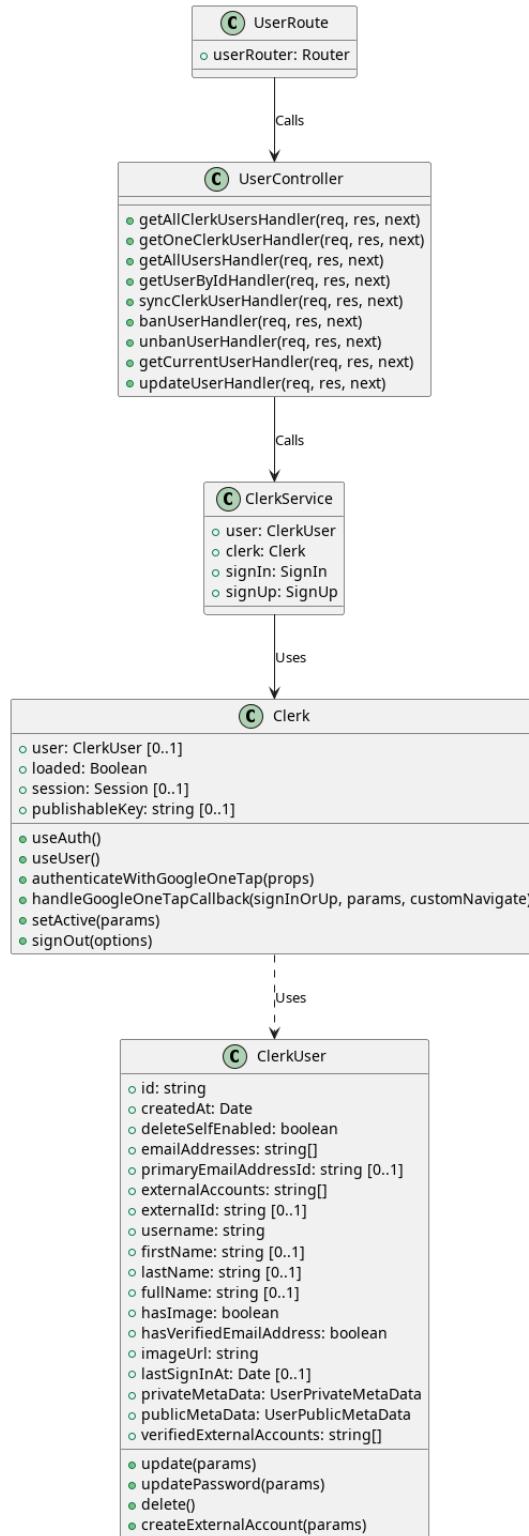


Figure 3.2.3.1: Update password information⁵¹

⁵¹ [update password.png](#)

3.2.3.2 Sequence diagram

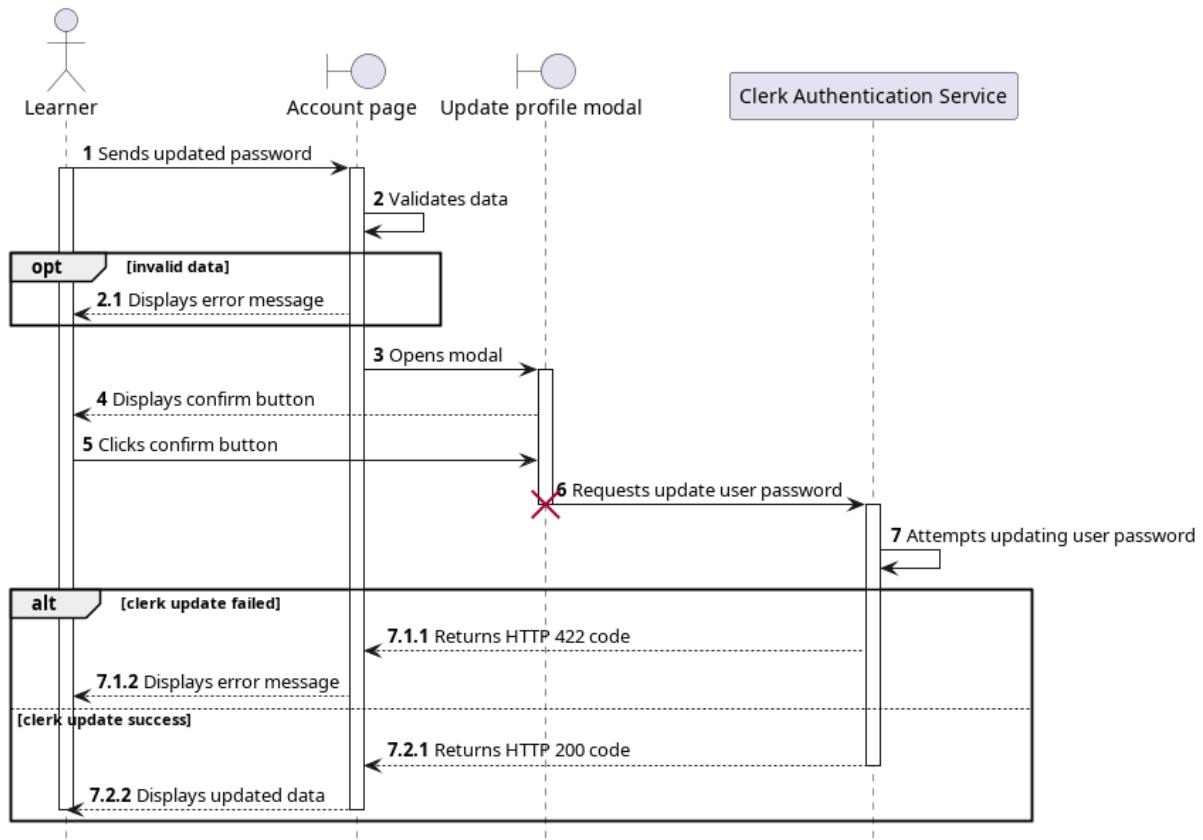


Figure 3.2.3.2.a: Learner update password information⁵²

⁵² [learner update profile data.png](#)

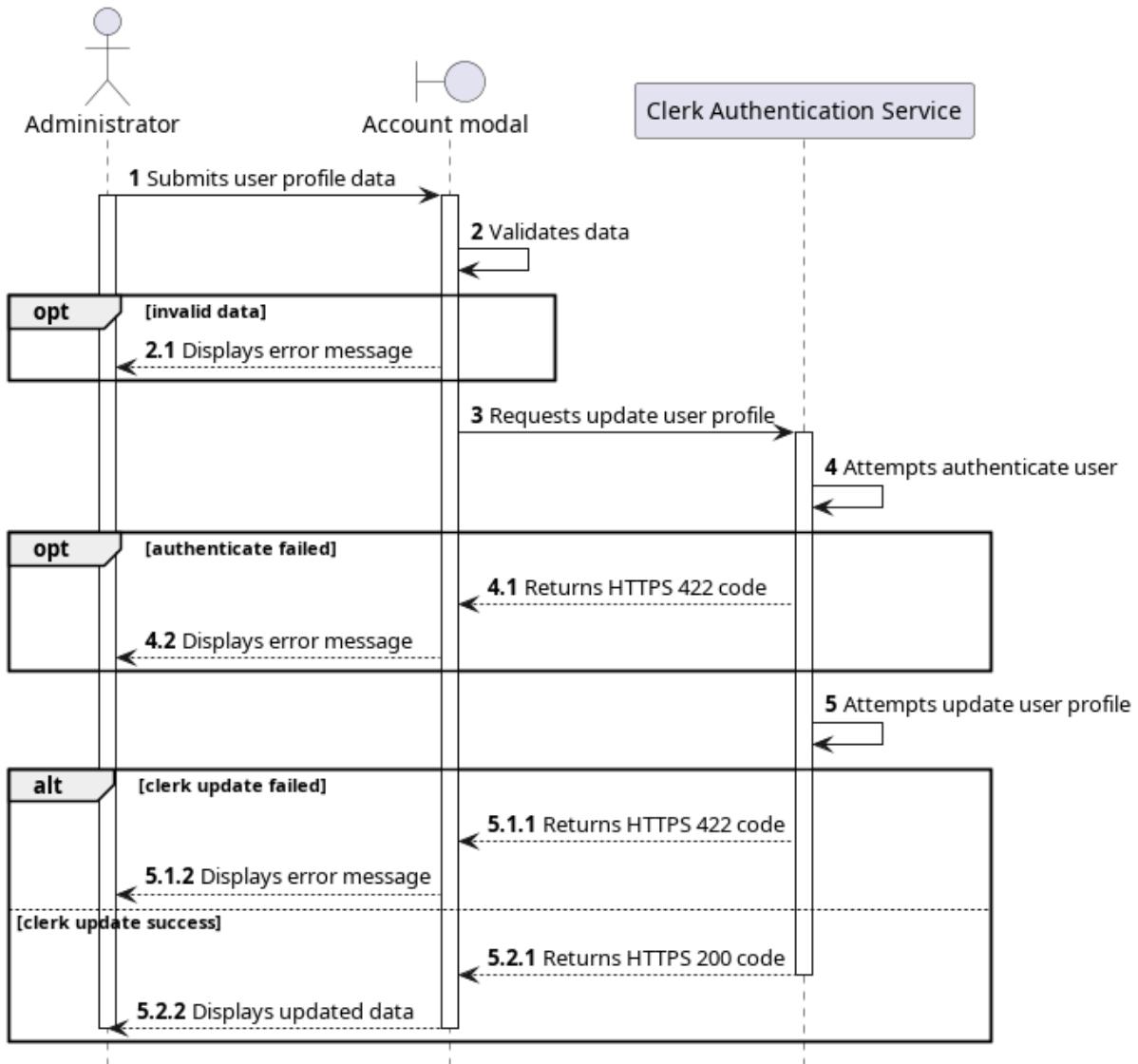


Figure 3.2.3.2.b: Administrator update password information⁵³

⁵³ [admin update password.png](#)

3.2.4 Reset password

3.2.4.1 Class diagram

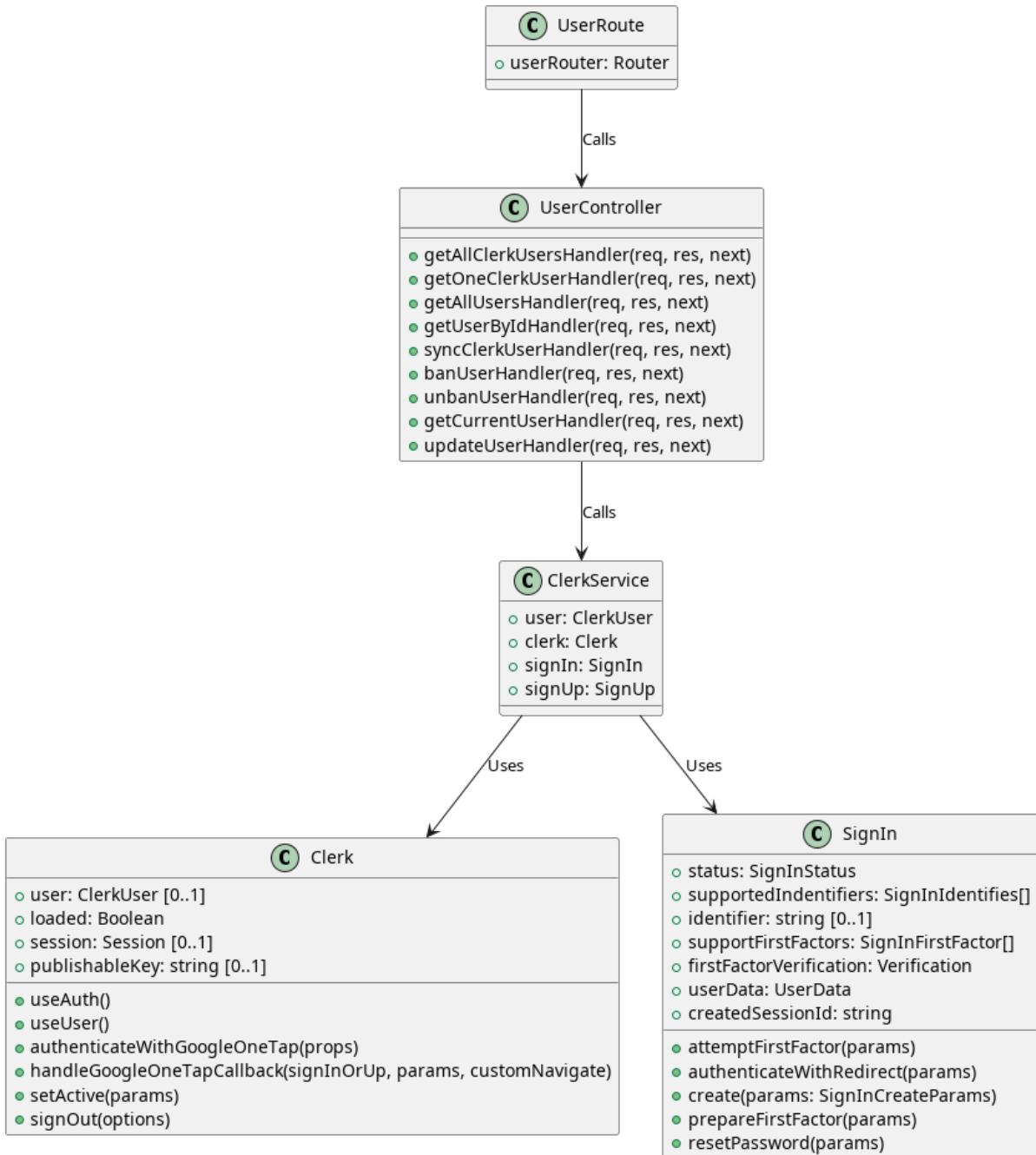


Figure 3.2.4.1: Reset password⁵⁴

⁵⁴ [admin update.png](#)

3.2.4.2 Sequence diagram

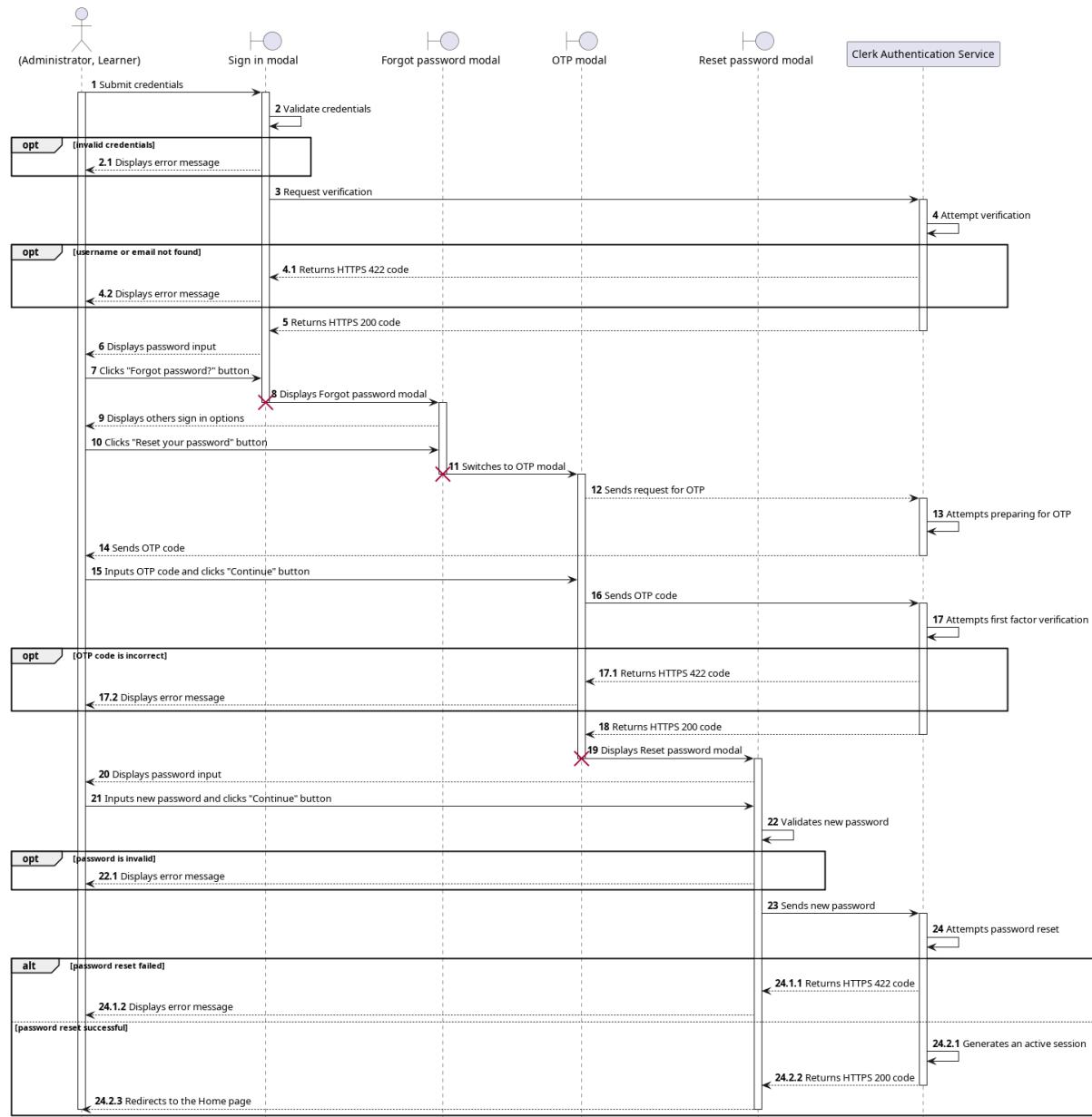


Figure 3.2.4.2: Reset password⁵⁵

⁵⁵ [reset password.png](#)

3.2.5 Delete account

3.2.5.1 Class diagram

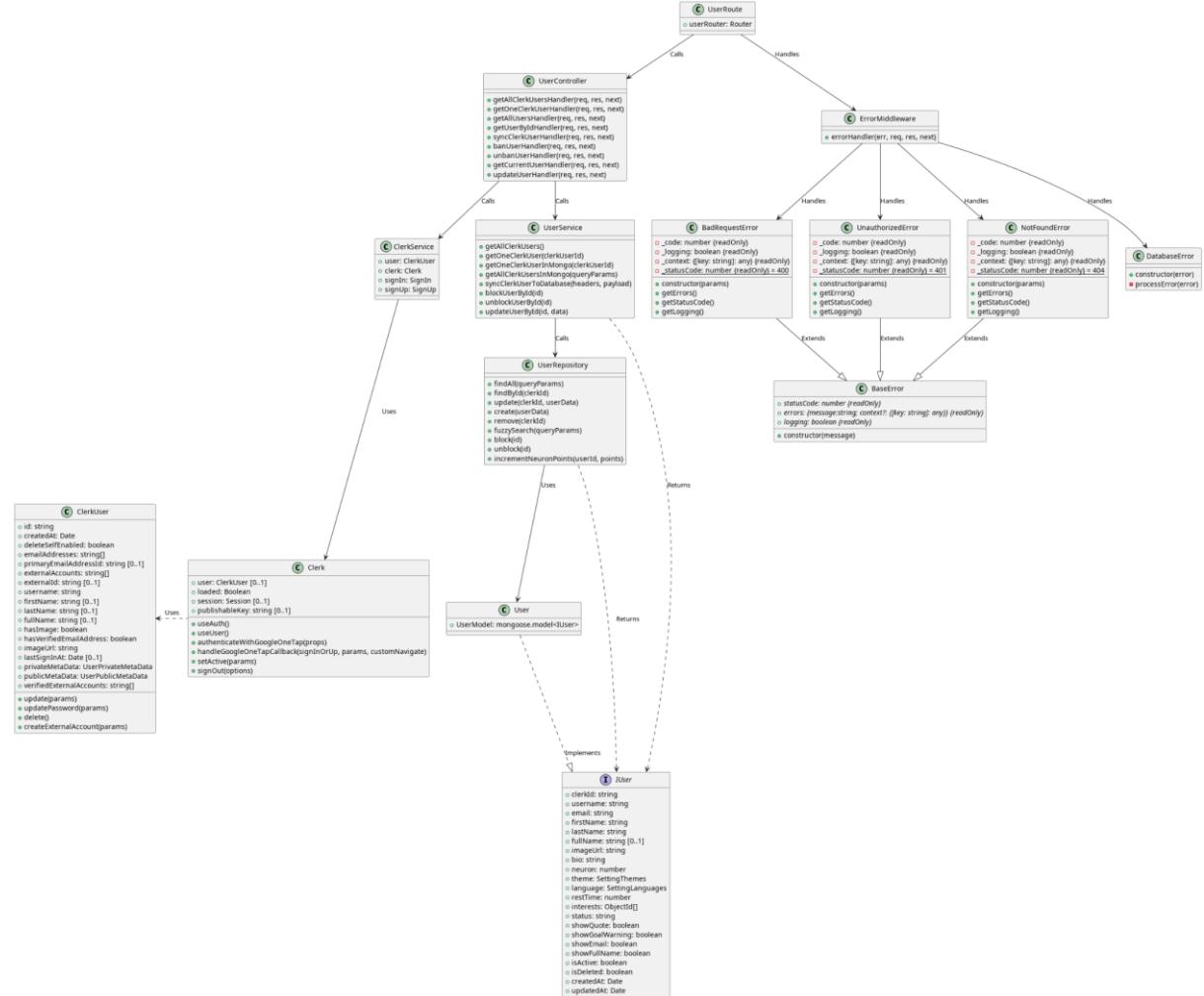


Figure 3.2.5.1: Delete account⁵⁶

⁵⁶ [sign up with credentials.png](#)

3.2.5.2 Sequence diagram

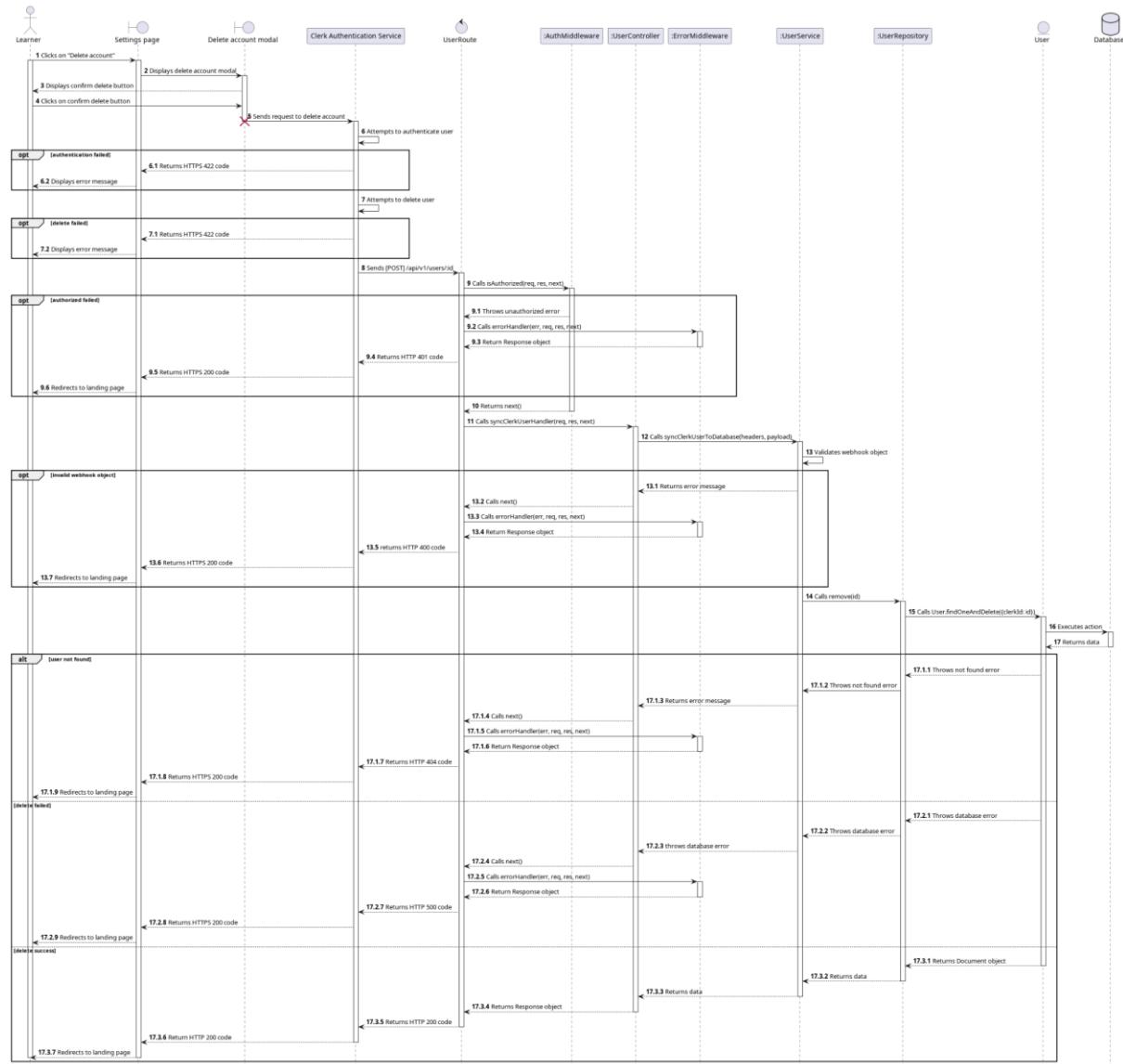


Figure 3.2.5.2: Delete account⁵⁷

⁵⁷ [delete account.png](#)

3.3 Question Set Management

3.3.1 View question sets

3.3.1.1 Class diagram

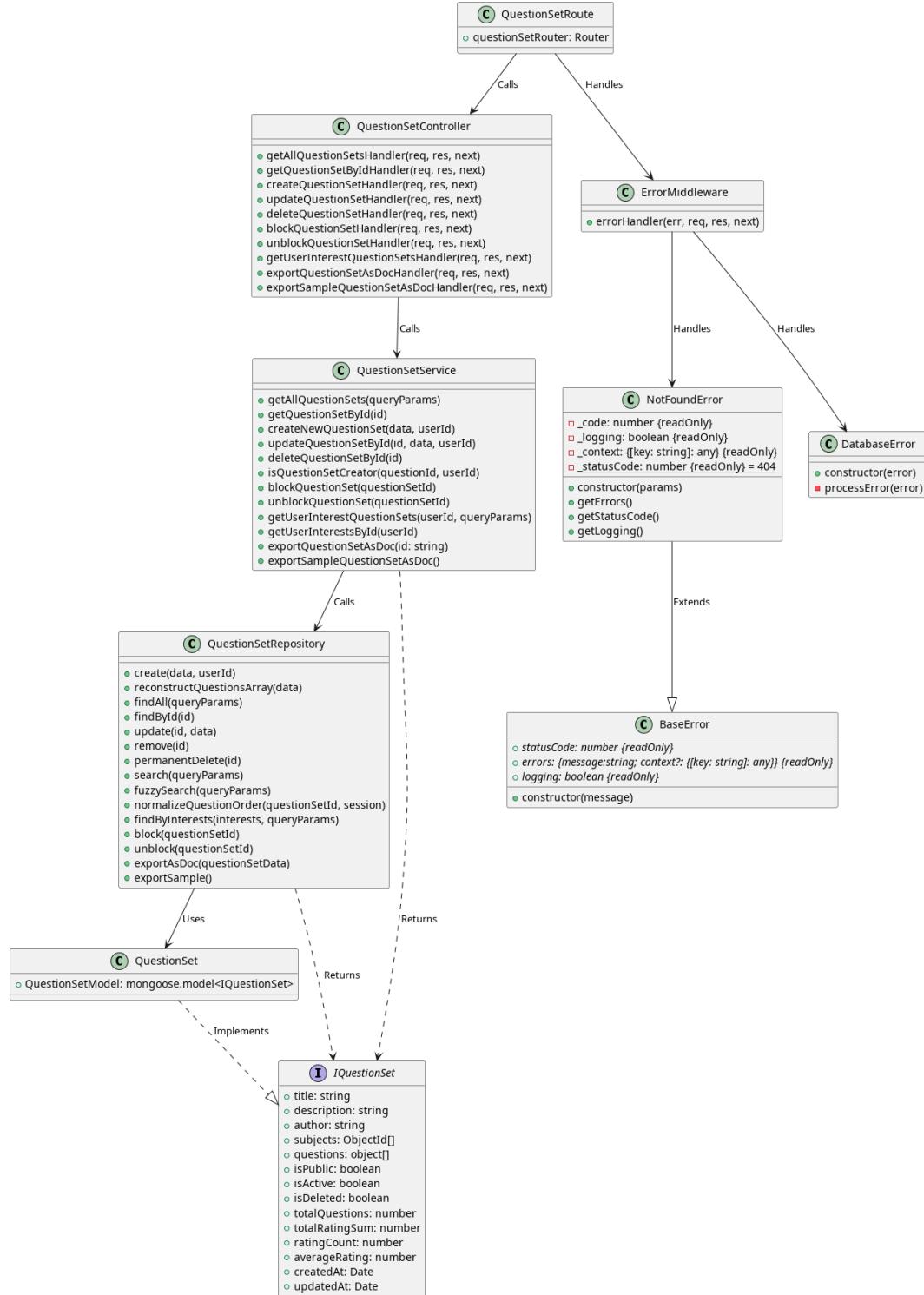


Figure 3.3.1.1: View question sets⁵⁸

⁵⁸ [view question sets.png](#)

3.3.1.2 Sequence diagram

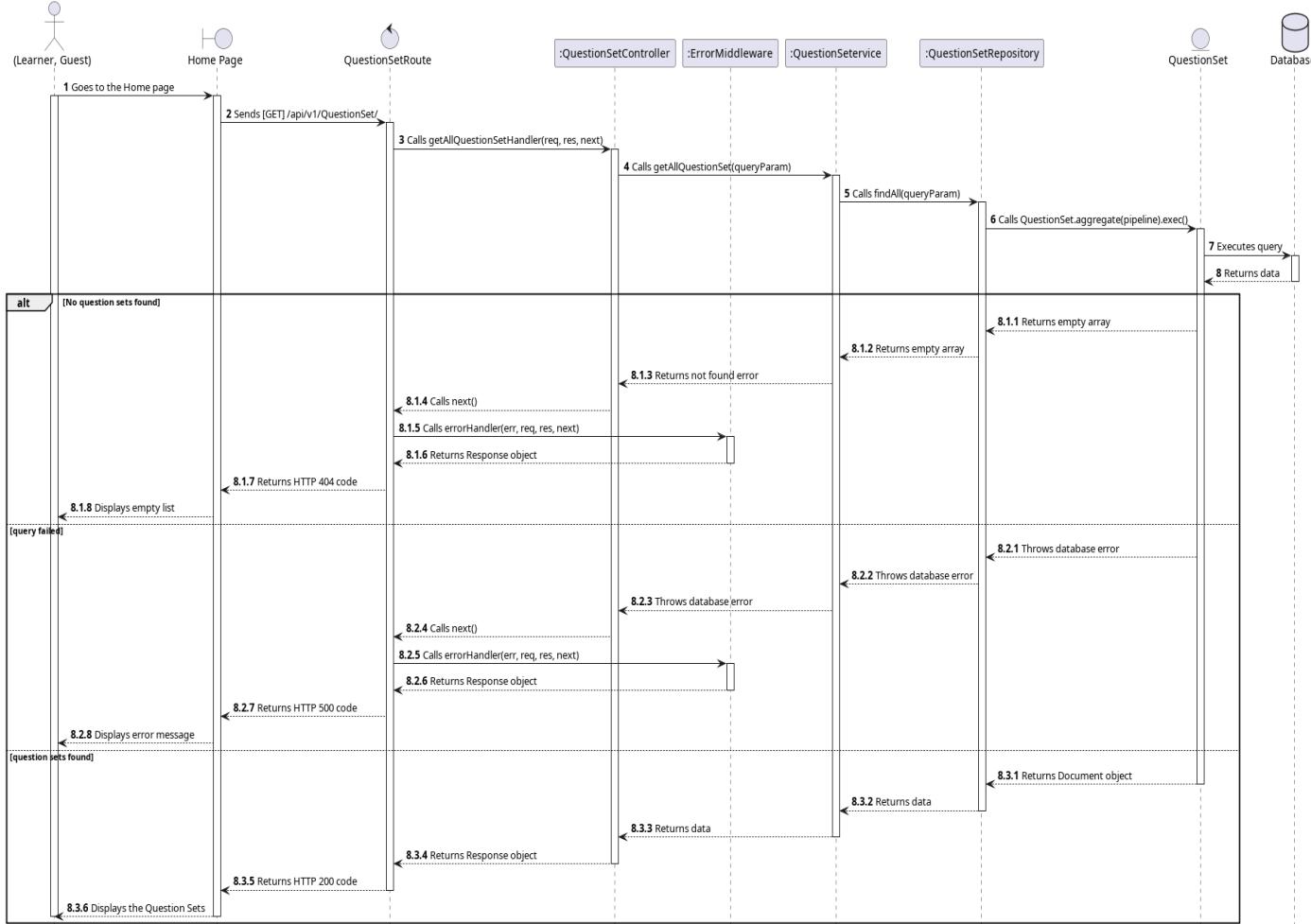


Figure 3.3.1.2.a: Learners and Guests view question sets⁵⁹

⁵⁹ [view question sets.png](#)

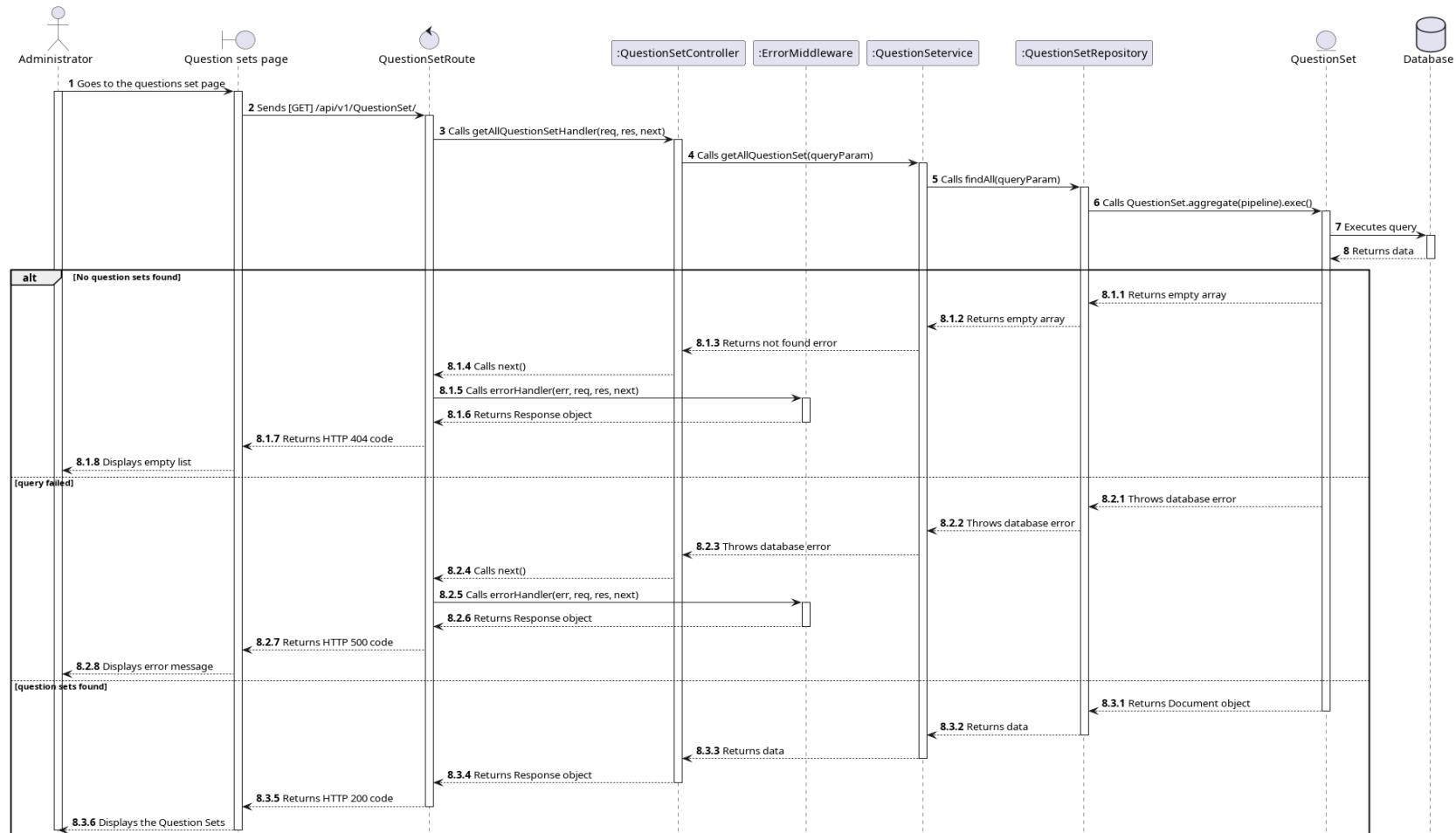


Figure 3.3.1.2.b: Administrator view question sets⁶⁰

⁶⁰ [view question sets.png](#)

3.3.2 View question set detail

3.3.2.1 Class diagram

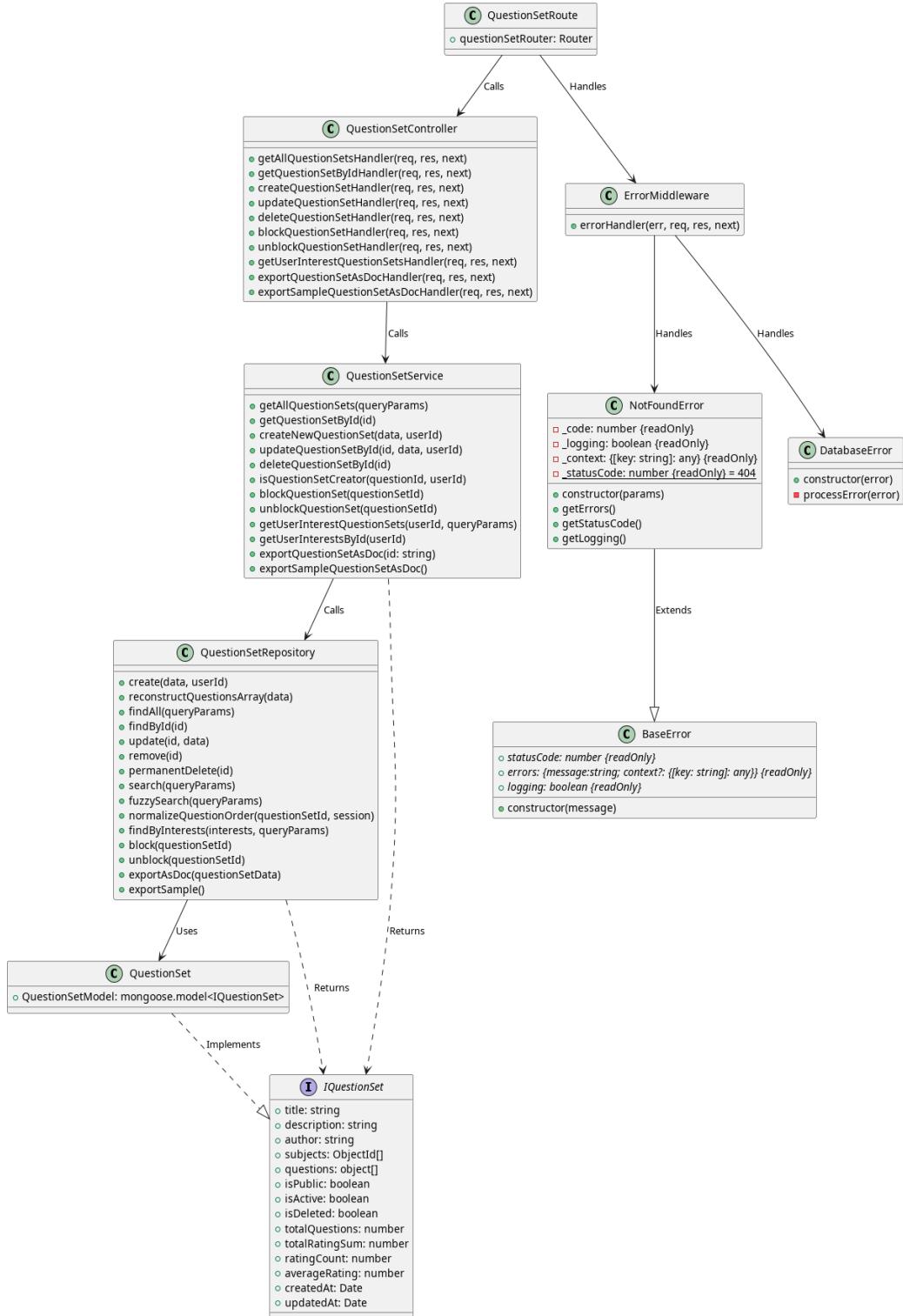


Figure 3.3.2.1: View question set detail⁶¹

⁶¹ [view question set details.png](#)

3.3.2.2 Sequence diagram

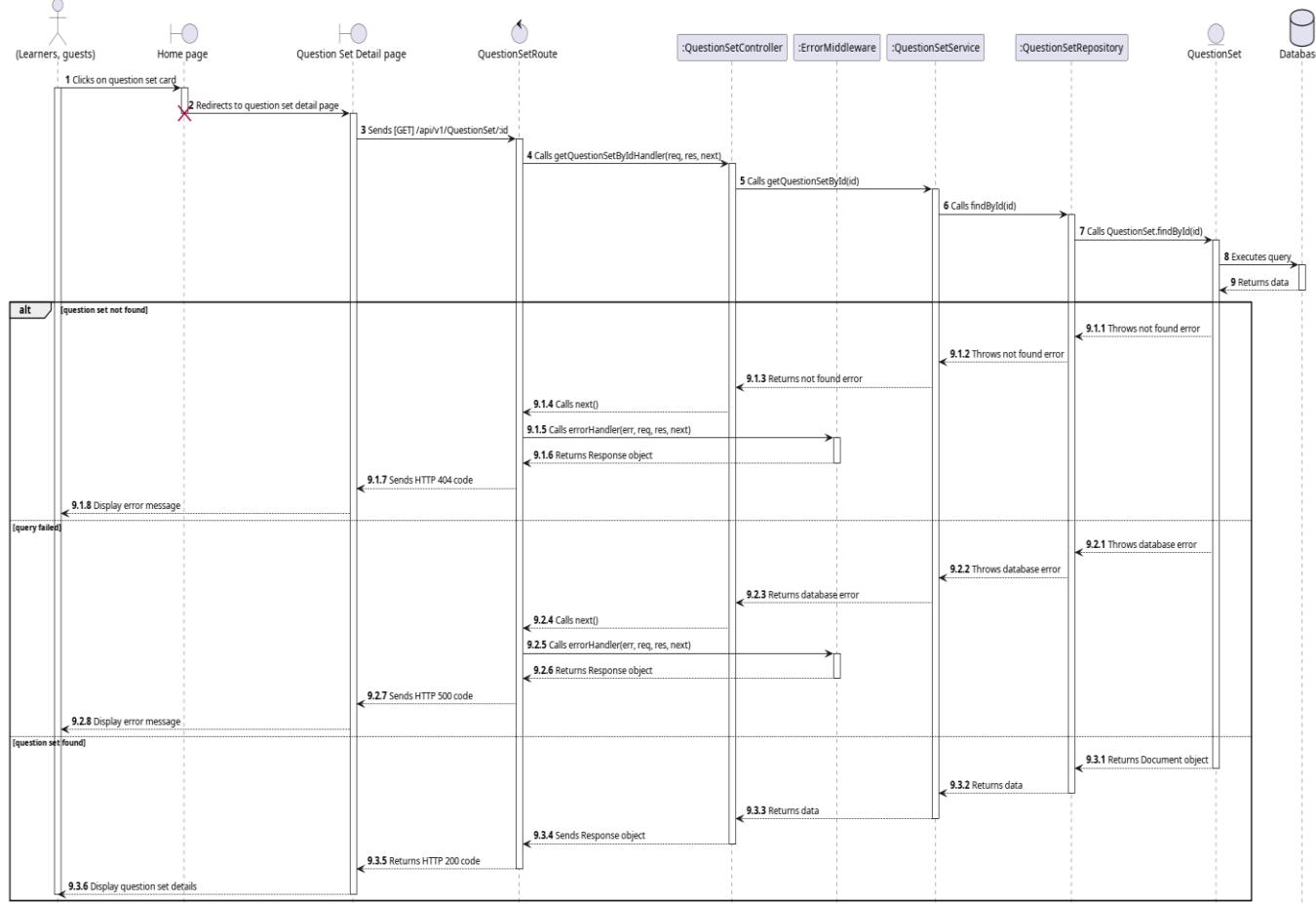


Figure 3.3.2.2.a: Learners and Guests view question set detail⁶²

⁶² [view question set details.png](#)

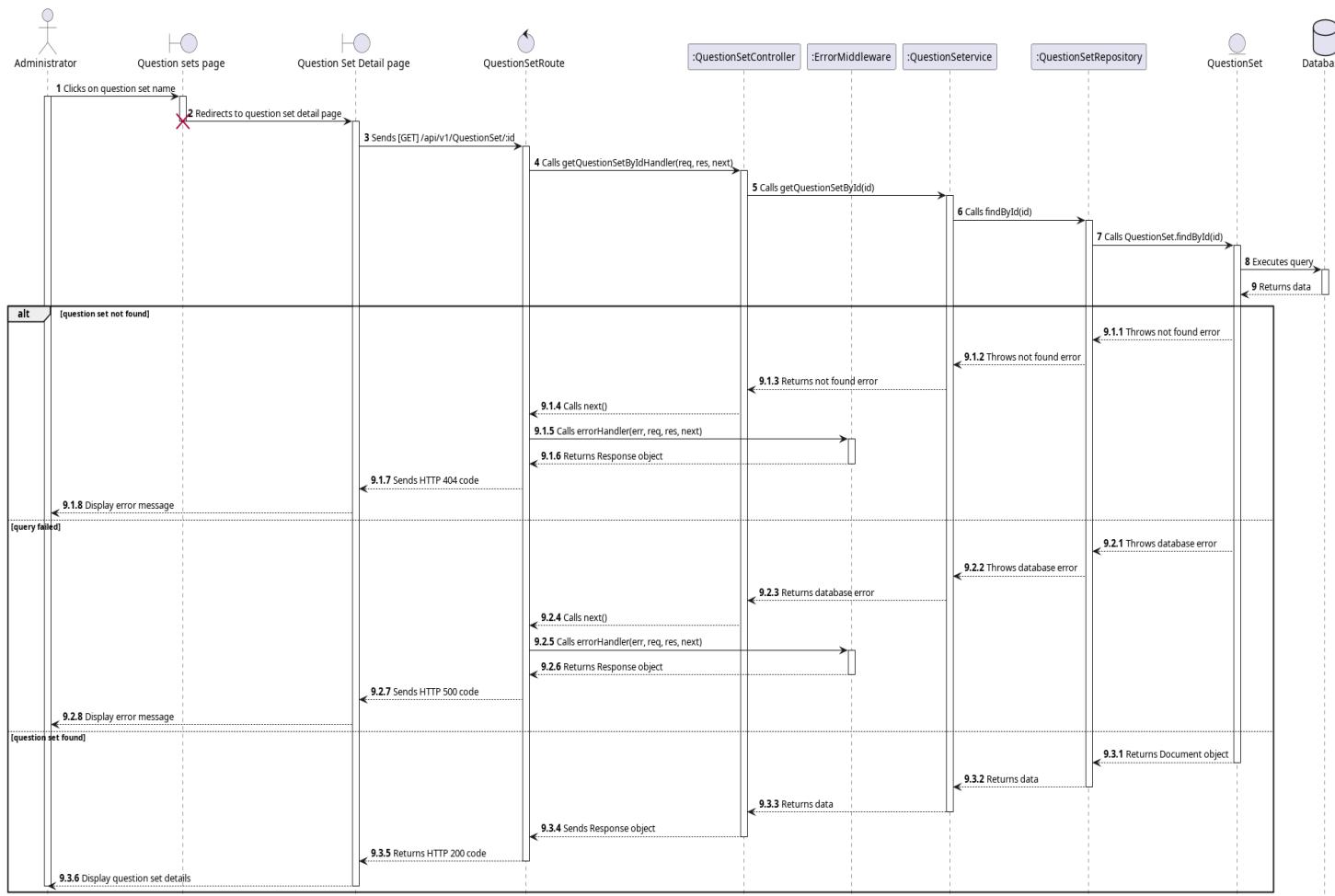


Figure 3.3.2.2.b: Administrator view question set detail⁶³

⁶³ [view question set details.png](#)

3.3.3 Create question sets

3.3.3.1 Class diagram

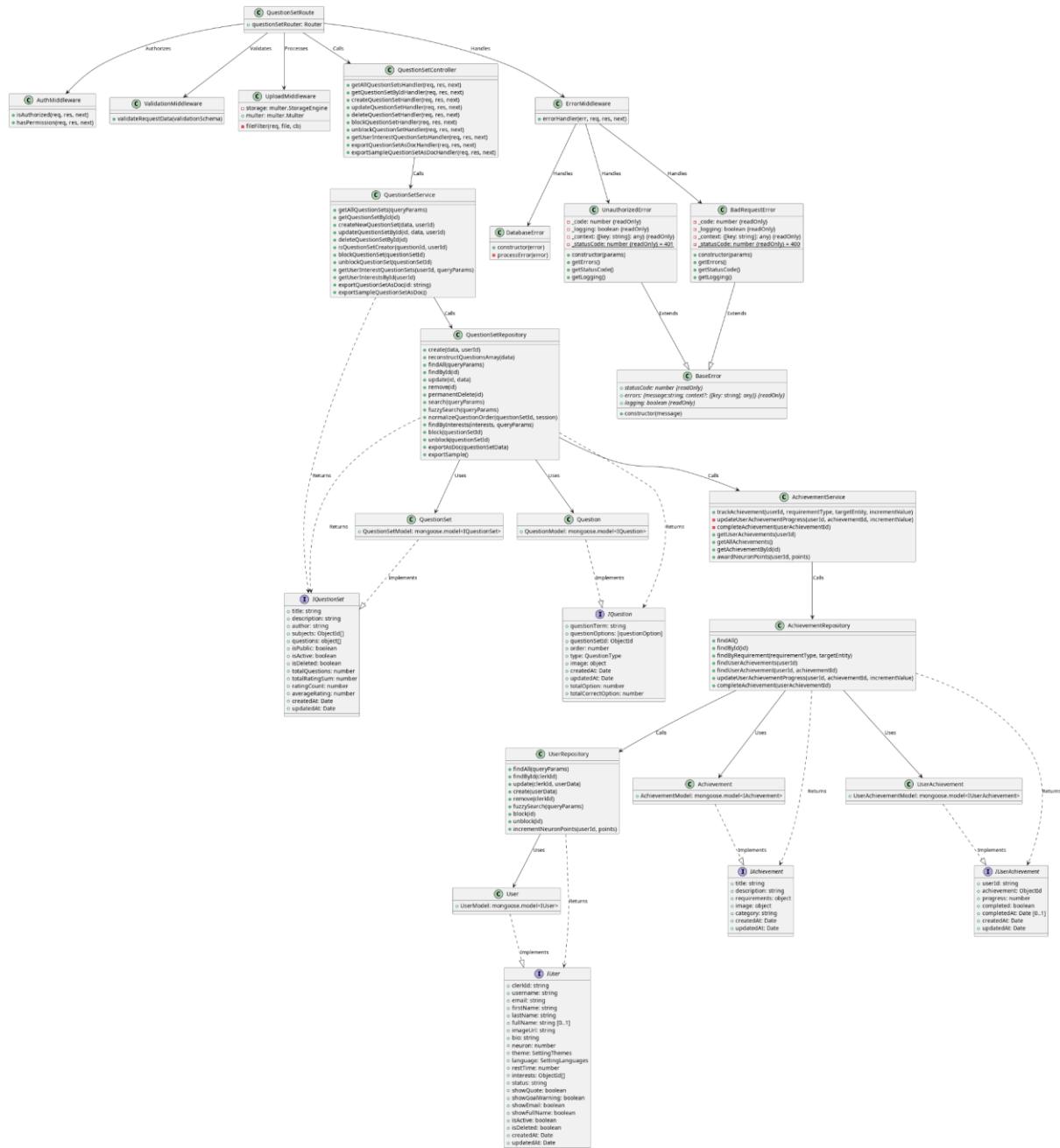


Figure 3.3.3.1: Create question set⁶⁴

⁶⁴ [create question set.png](#)

3.3.3.2 Sequence diagrams

3.3.3.2.1 Create question set

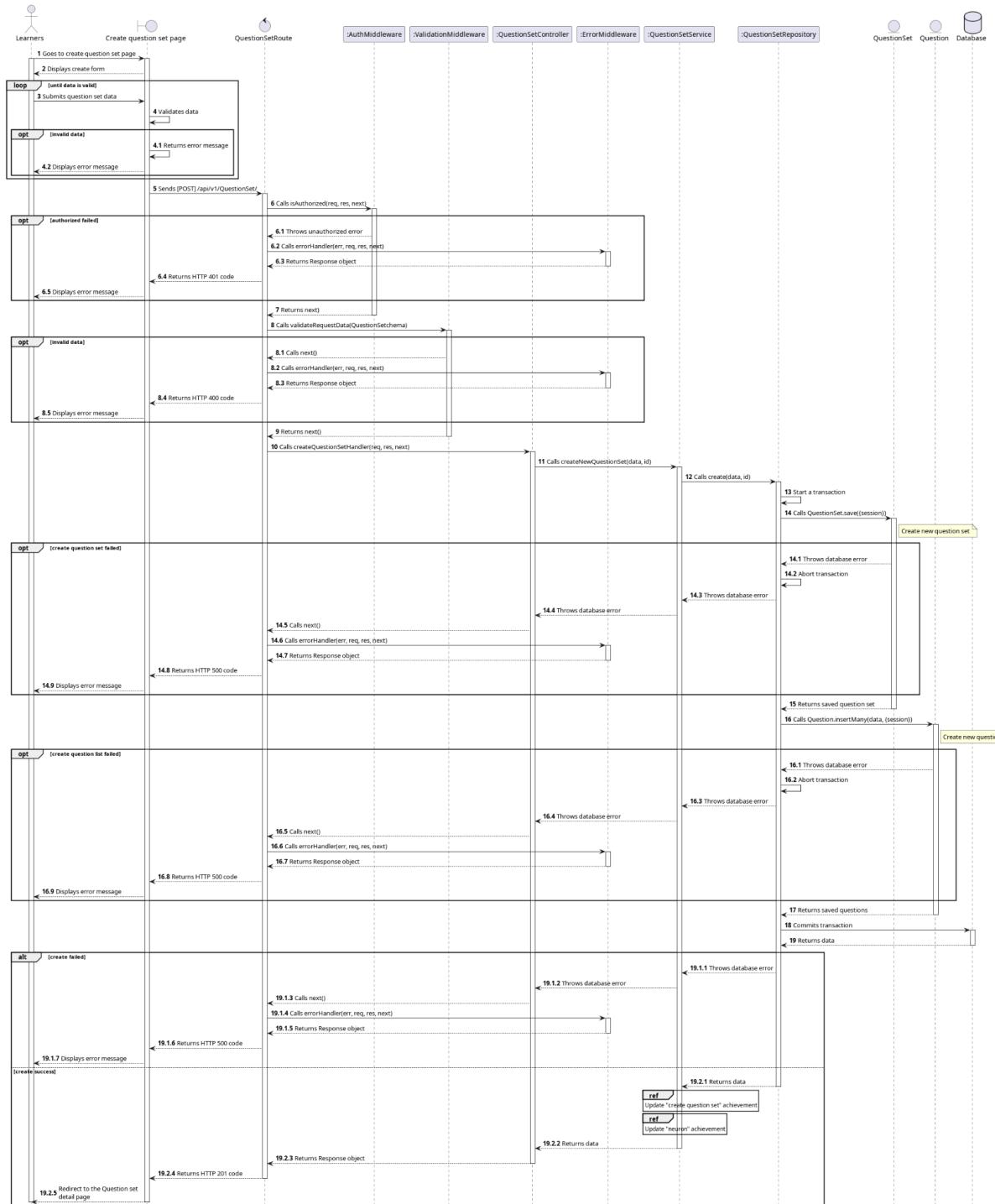


Figure 3.3.3.2.1: Create question set⁶⁵

⁶⁵ [create question set.png](#)

3.3.3.2.2 Update “create question set” achievement

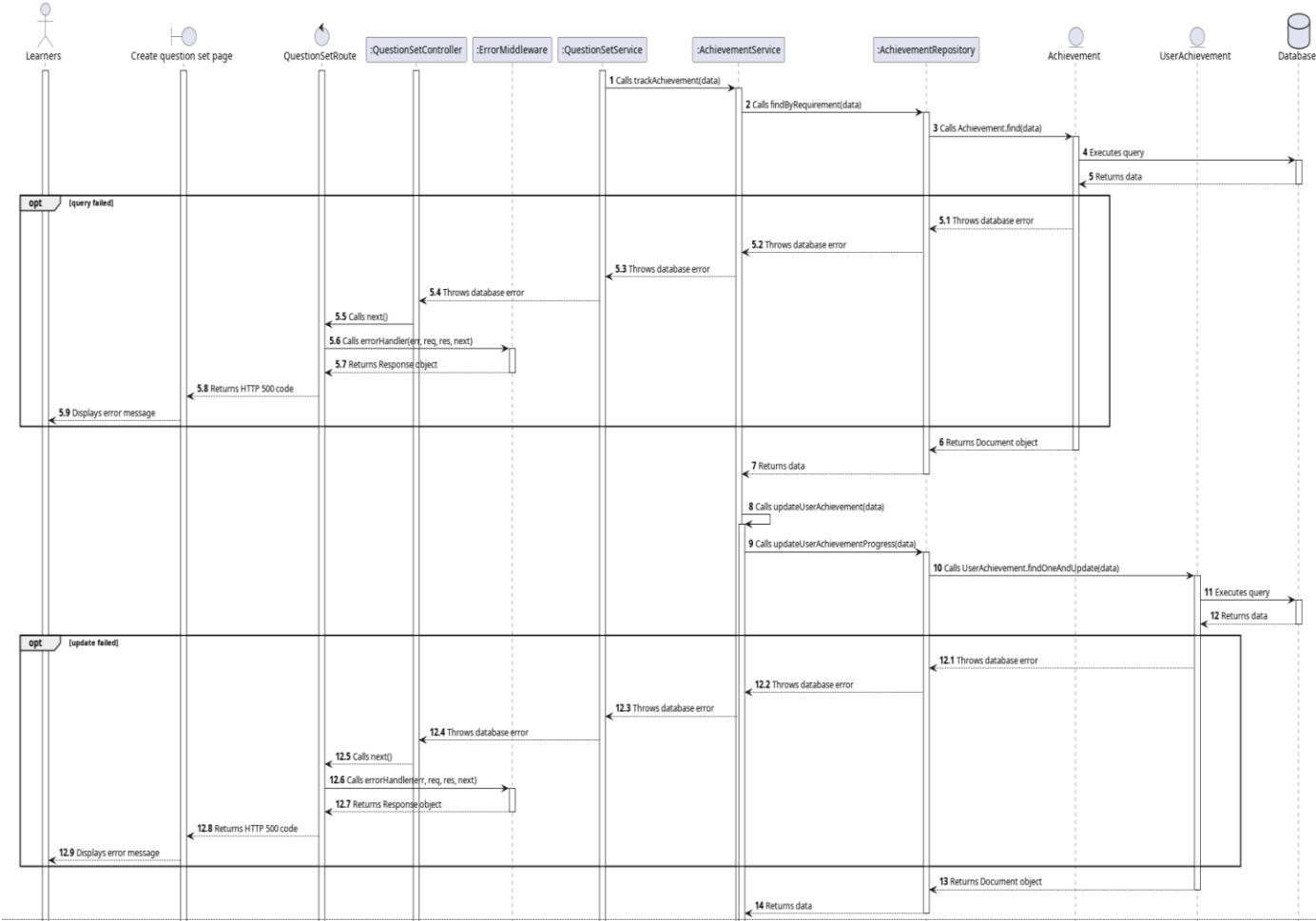


Figure 3.3.3.2.2.a: Update “create question set” achievement - 1⁶⁶

⁶⁶ [create question set achievement.png](#)

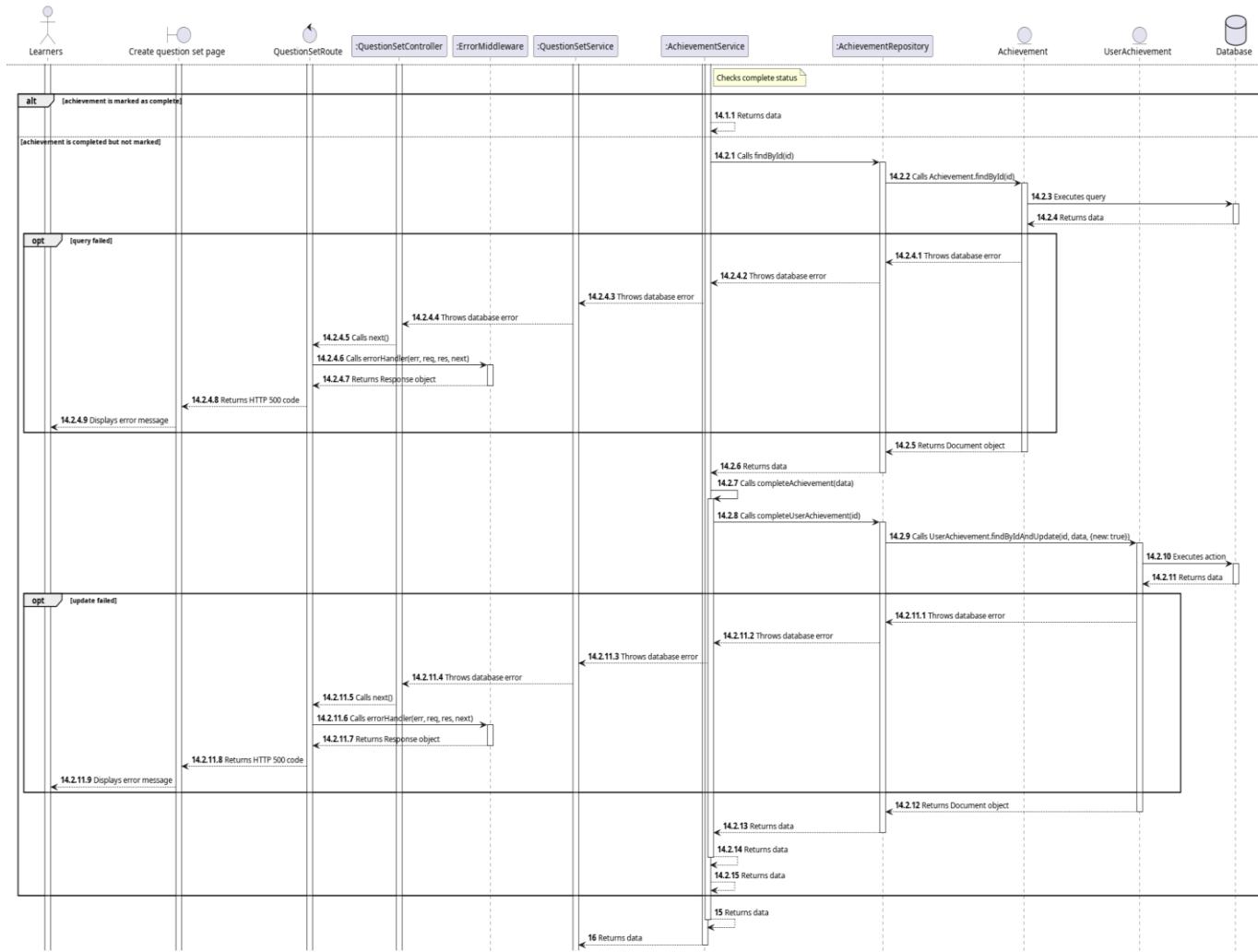


Figure 3.3.3.2.2.b: Update “create question set” achievement - 2

3.3.3.2.3 Update “neuron” achievement

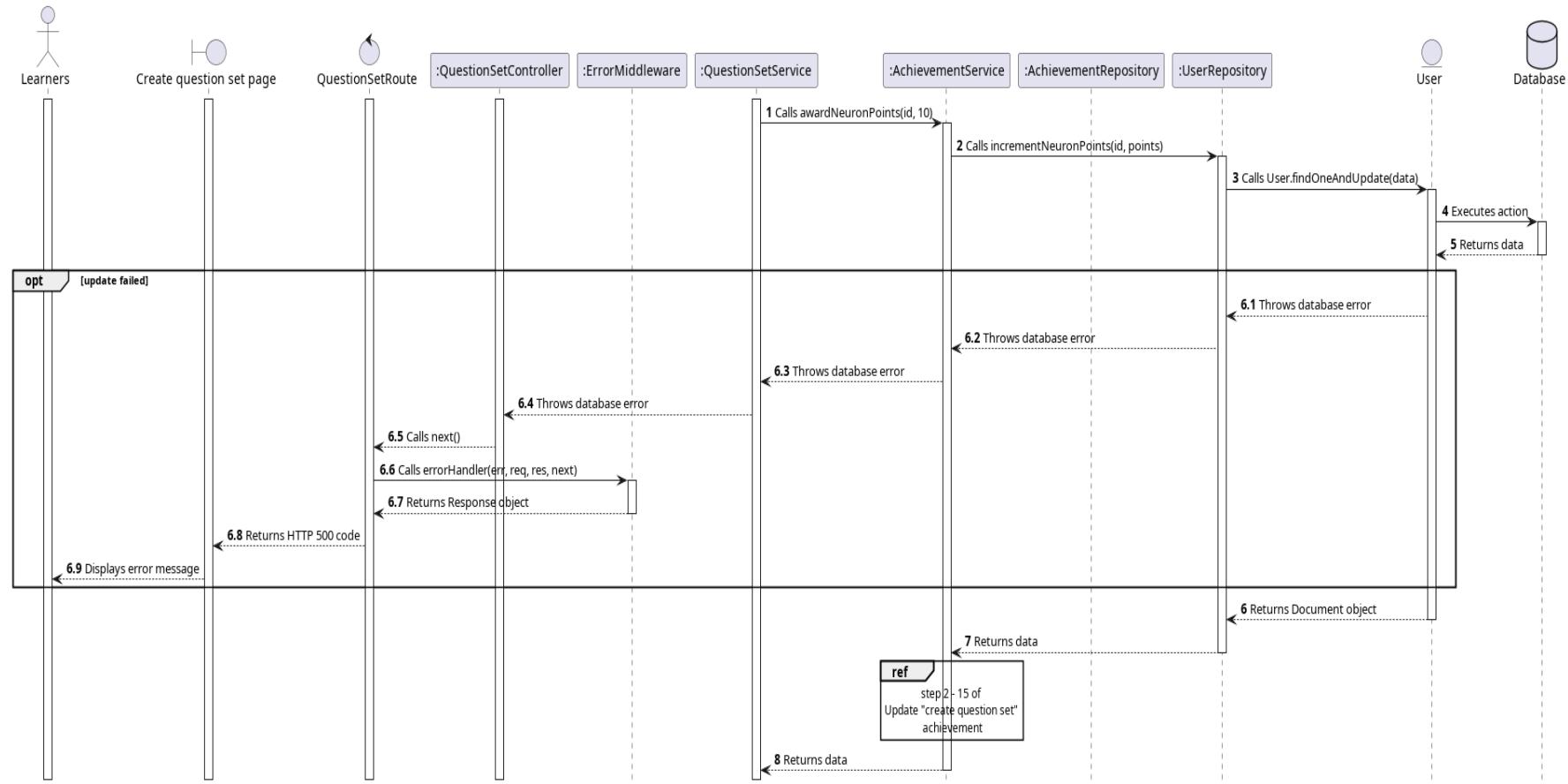


Figure 3.3.3.2.3: Update “neuron” achievement⁶⁷

⁶⁷ [neuron achievement.png](#)

3.3.4 Update question sets

3.3.4.1 Class diagram

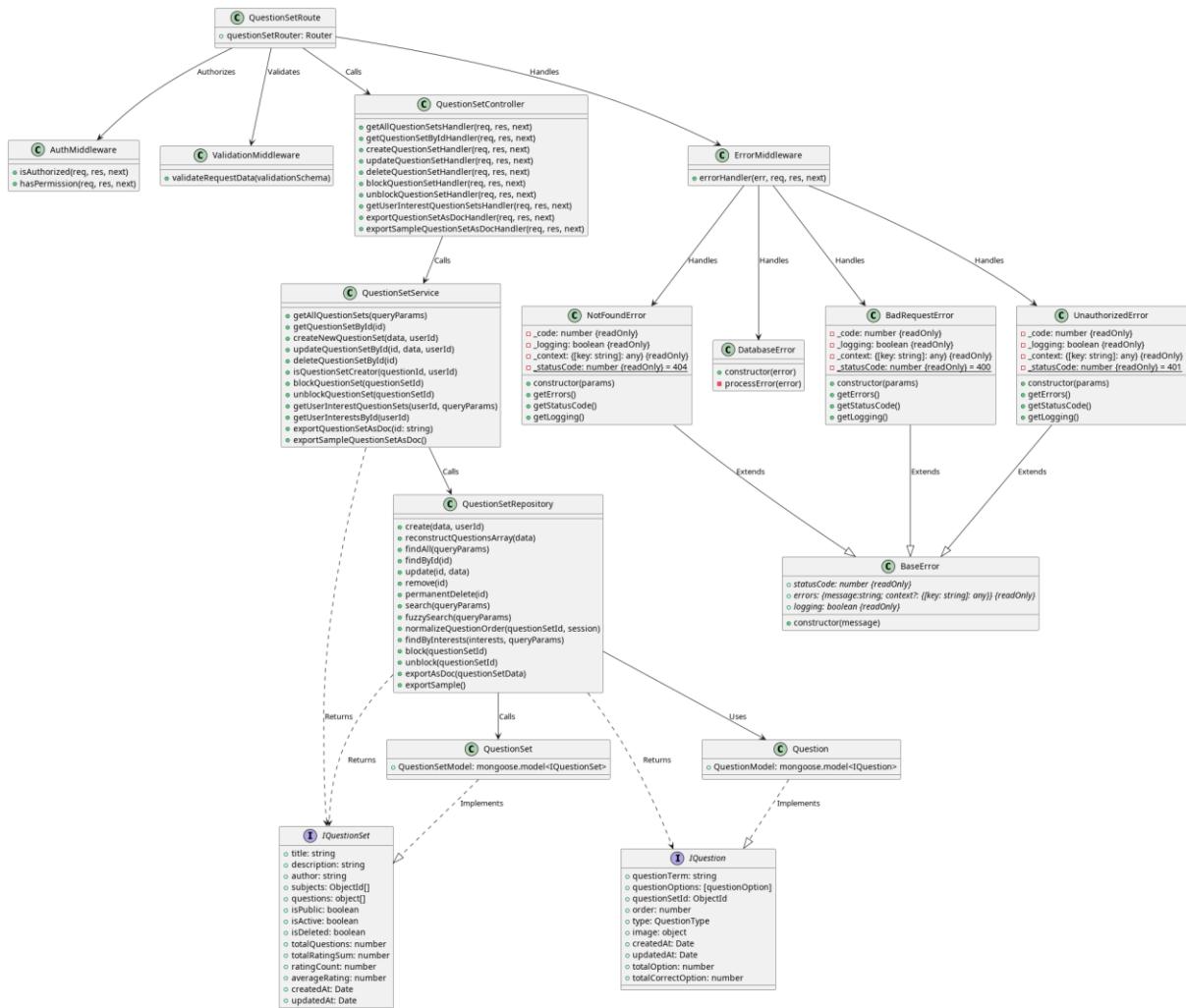


Figure 3.3.4.1: Update question set⁶⁸

⁶⁸ [update question set.png](#)

3.3.4.2 Sequence diagrams

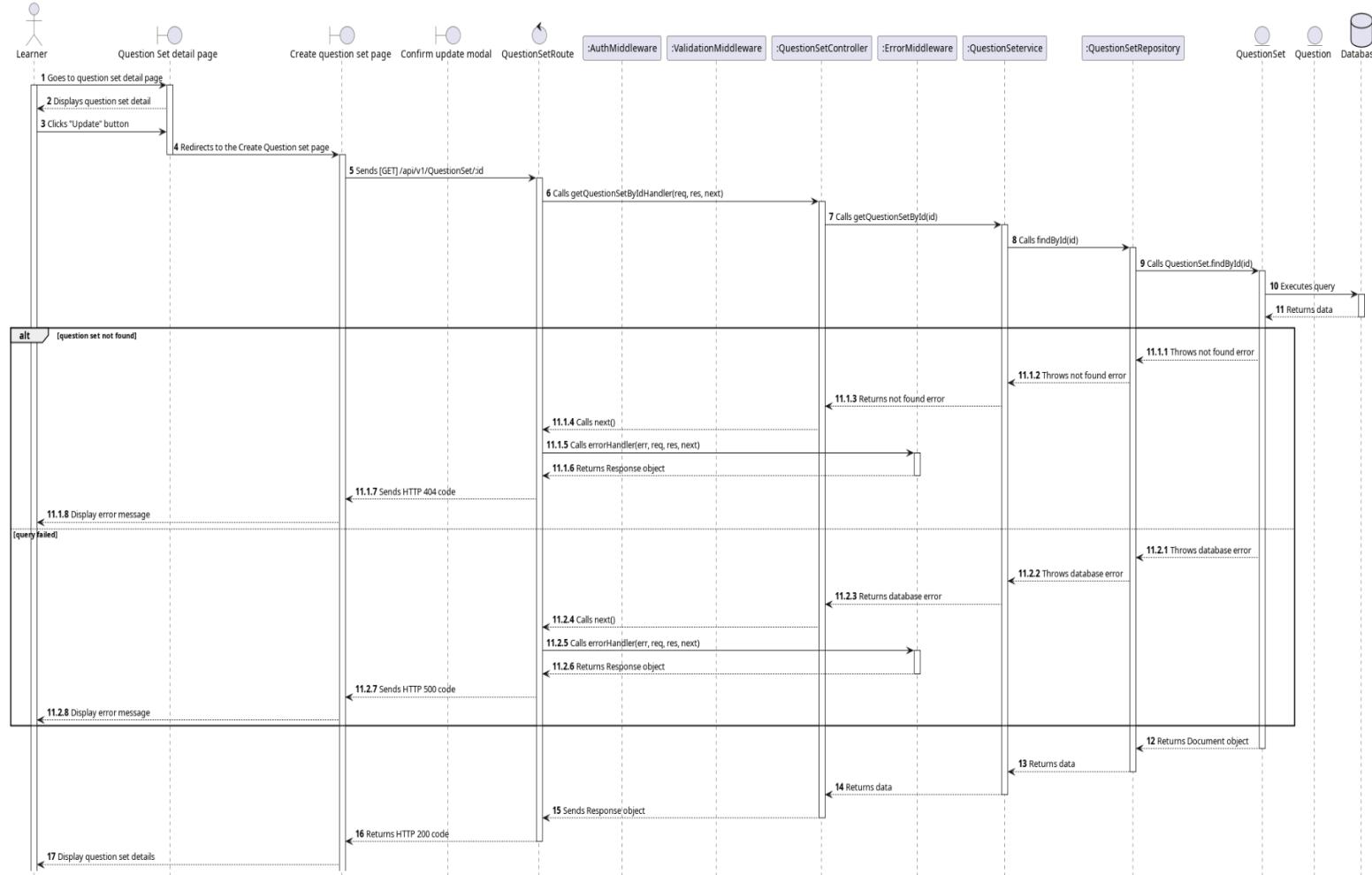


Figure 3.3.4.2.a: Get question set detail⁶⁹

⁶⁹ [view question set details.png](#)

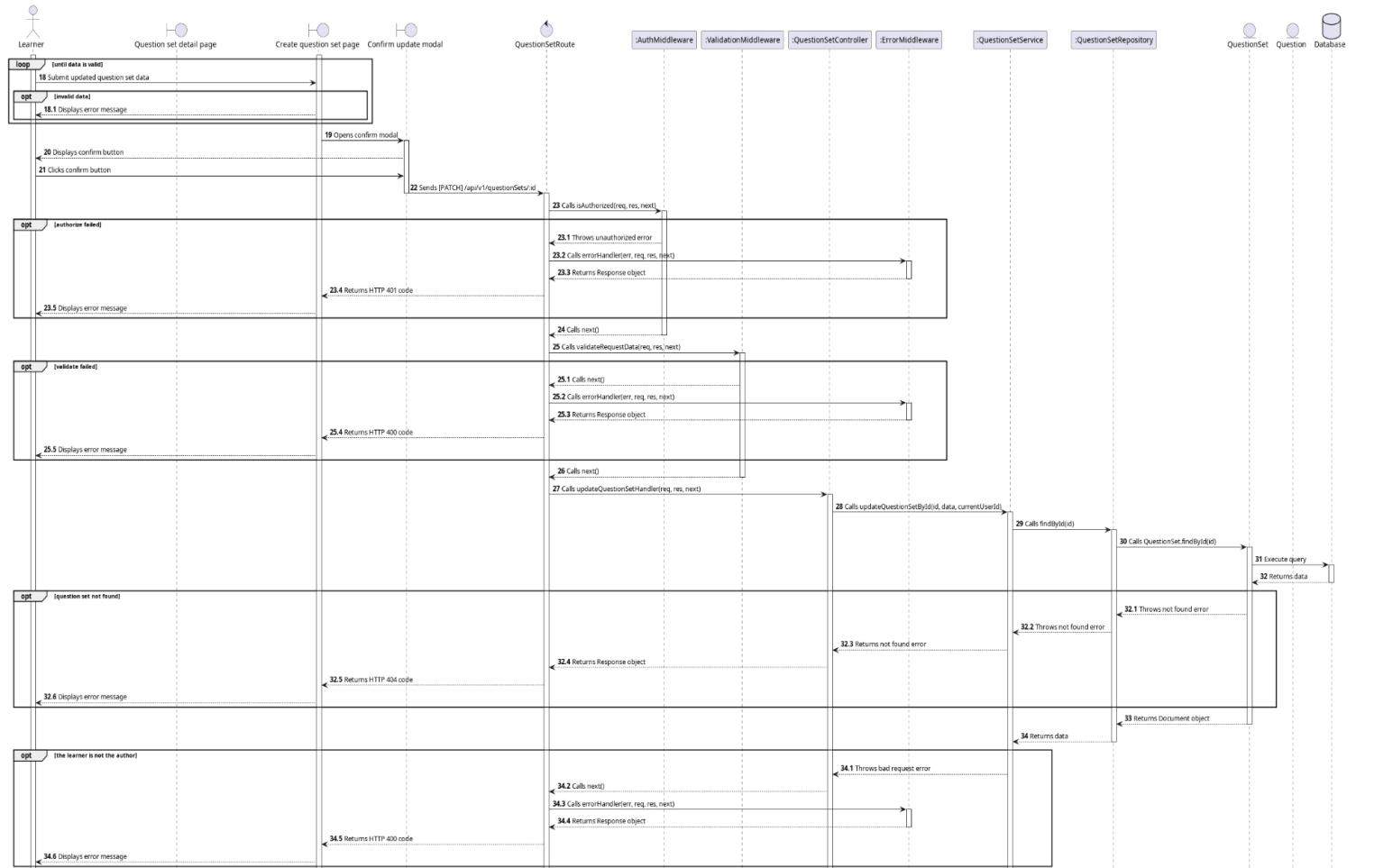


Figure 3.3.4.2.b: Update question set - 1⁷⁰

⁷⁰ [update question set.png](#)

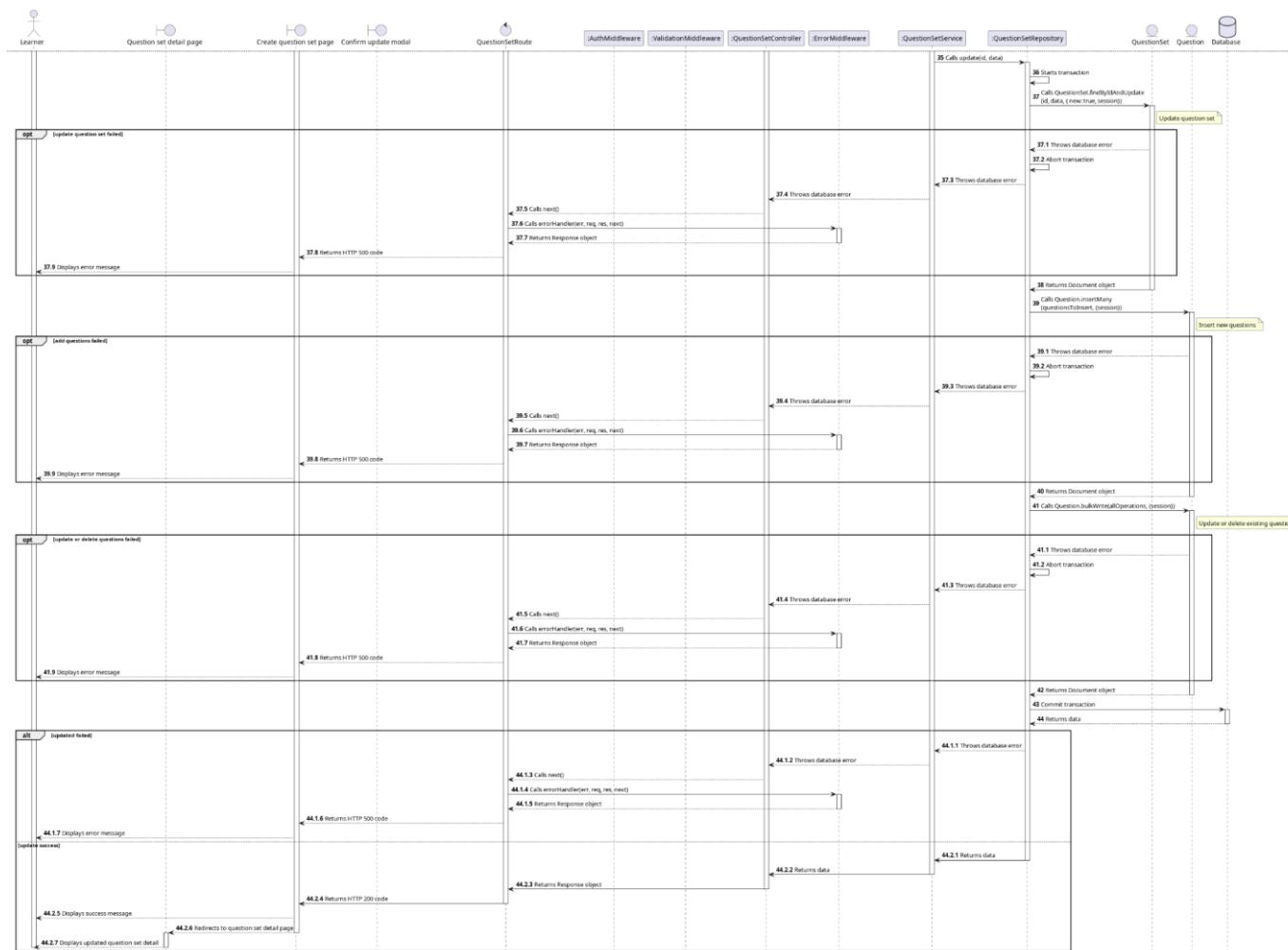


Figure 3.3.4.2.c: Update question set - 2

3.3.5 Delete question sets

3.3.5.1 Class diagram

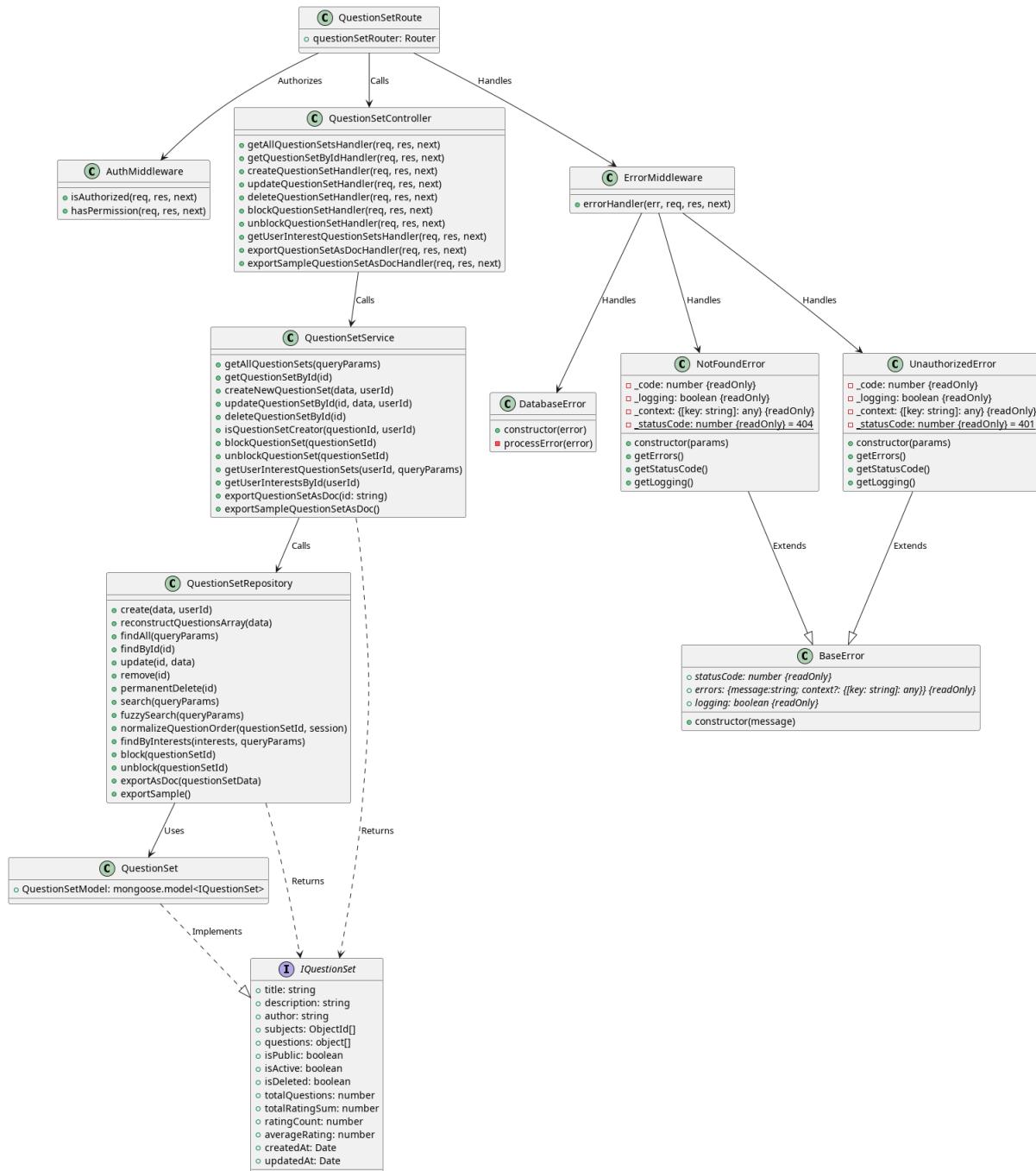


Figure 3.3.5.1: Delete question set⁷¹

⁷¹ [delete question set.png](#)

3.3.5.2 Sequence diagram

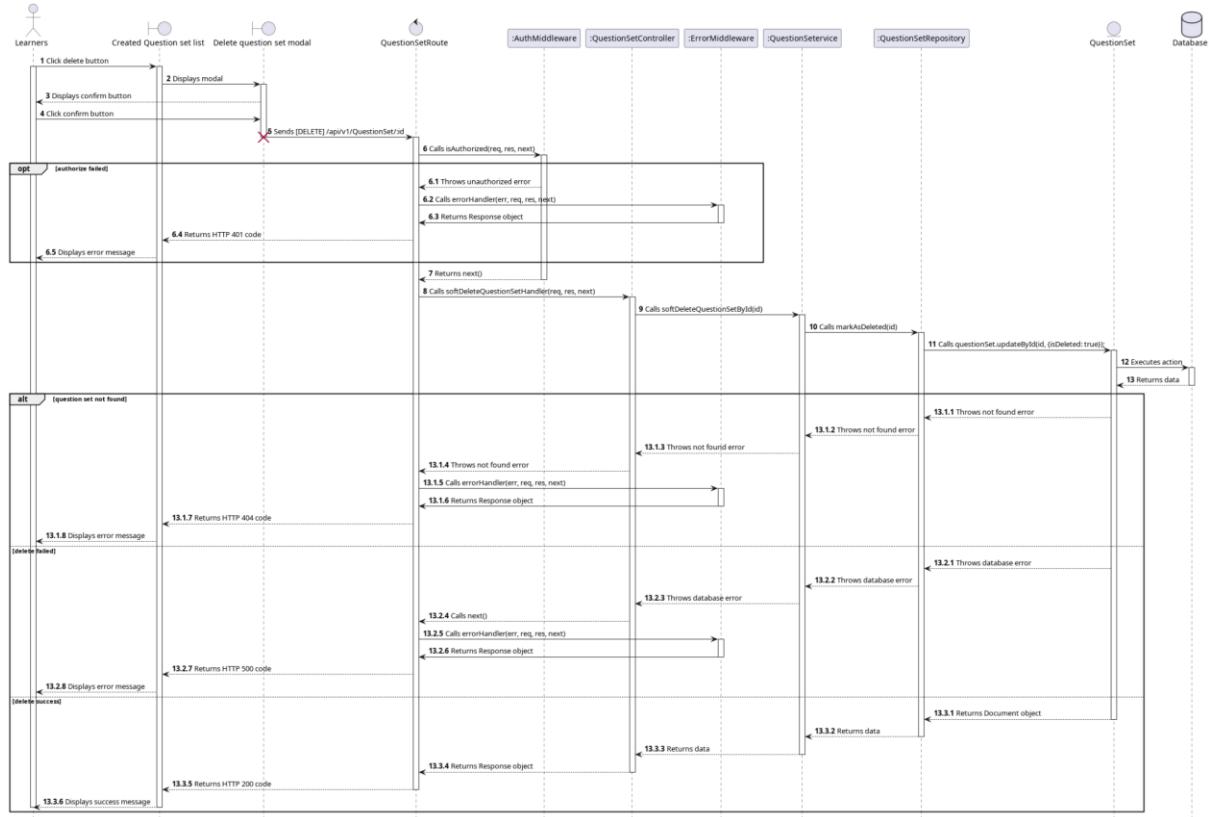


Figure 3.3.5.2: Delete question set⁷²

⁷² [delete question set.png](#)

3.3.6 Share question sets

3.3.6.1 Class diagram

3.3.6.1 Share question set with URL

* This feature is client only, so there will be no class diagram *

3.3.6.2 Share question set to group

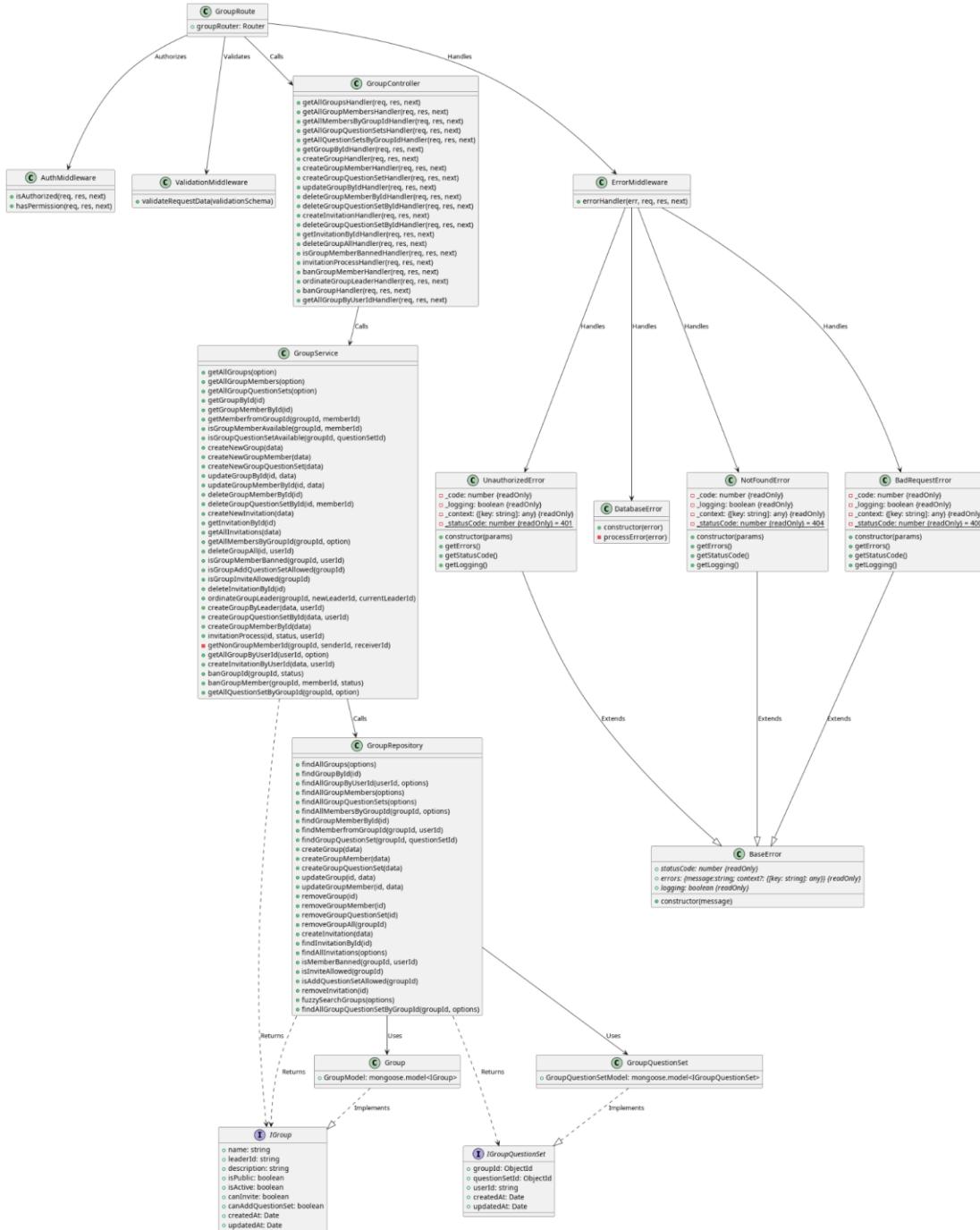


Figure 3.3.6.2: Share question set⁷³

⁷³ share question set to group.png

3.3.6.2 Sequence diagram

3.3.6.2.1 Share question set with URL

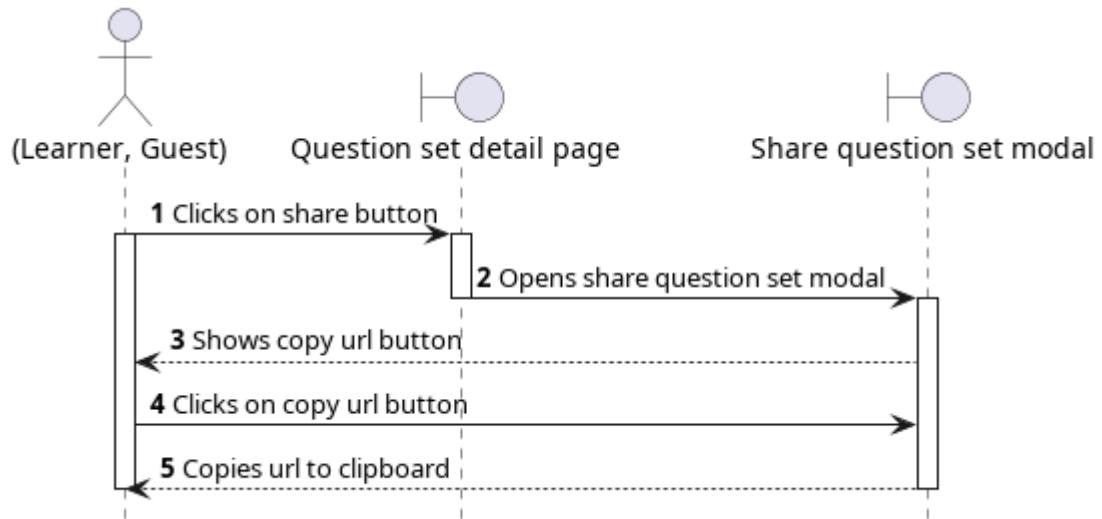


Figure 3.3.6.2.1: Share question set with URL

3.3.6.2.2 Share question set to group

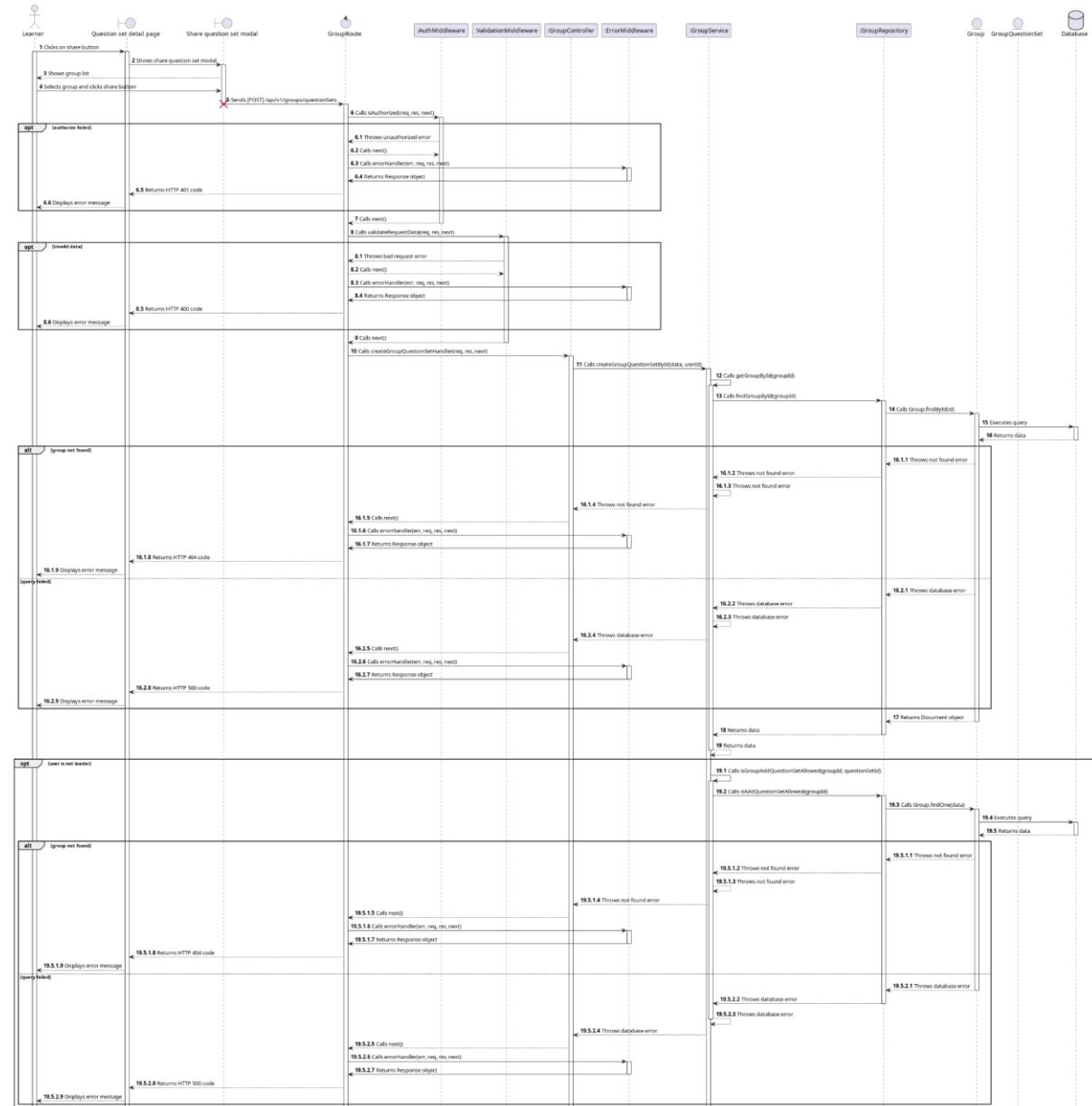


Figure 3.3.6.2.2.a: Share question set to group - 1⁷⁴

⁷⁴ [share question set to group.png](#)

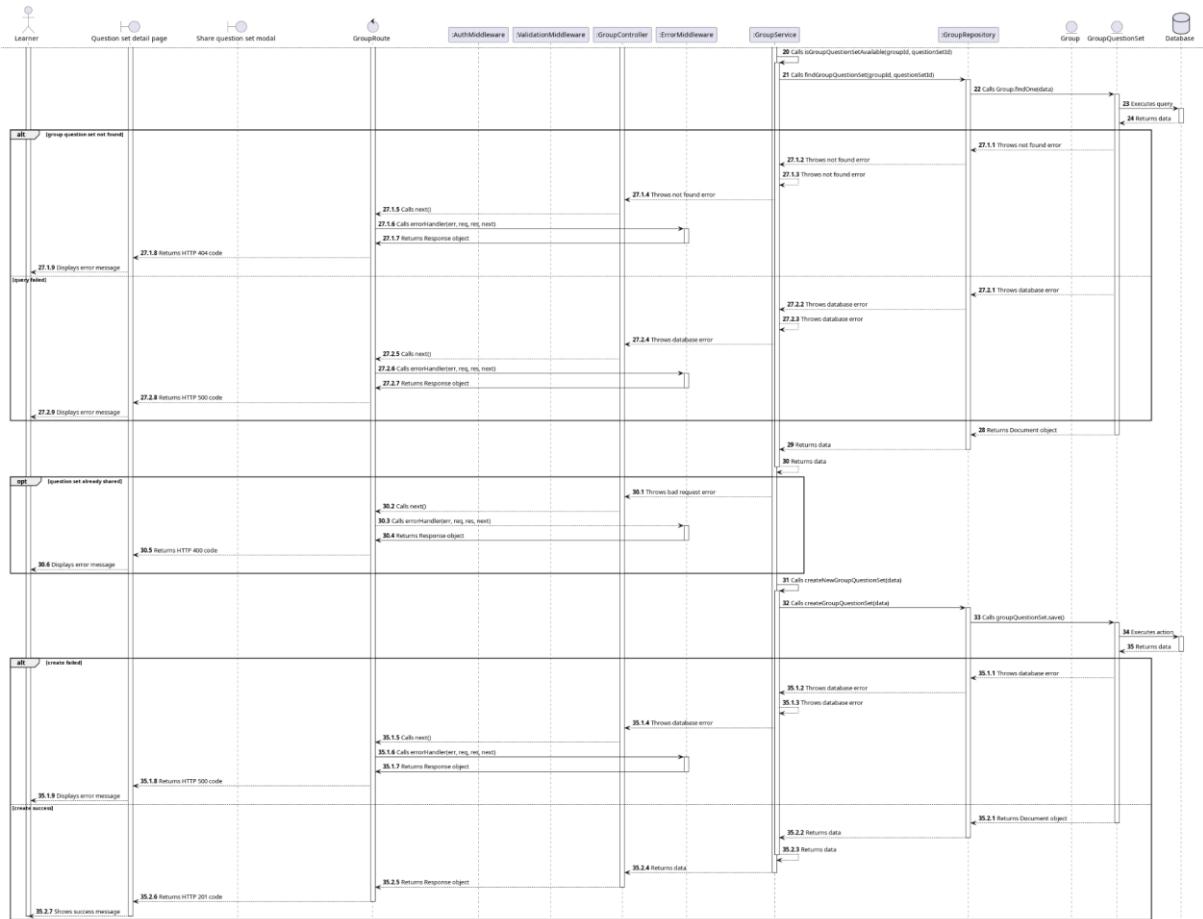


Figure 3.3.6.2.2.b: Share question set to group -2

3.3.7 Export question sets

3.3.7.1 Class diagram

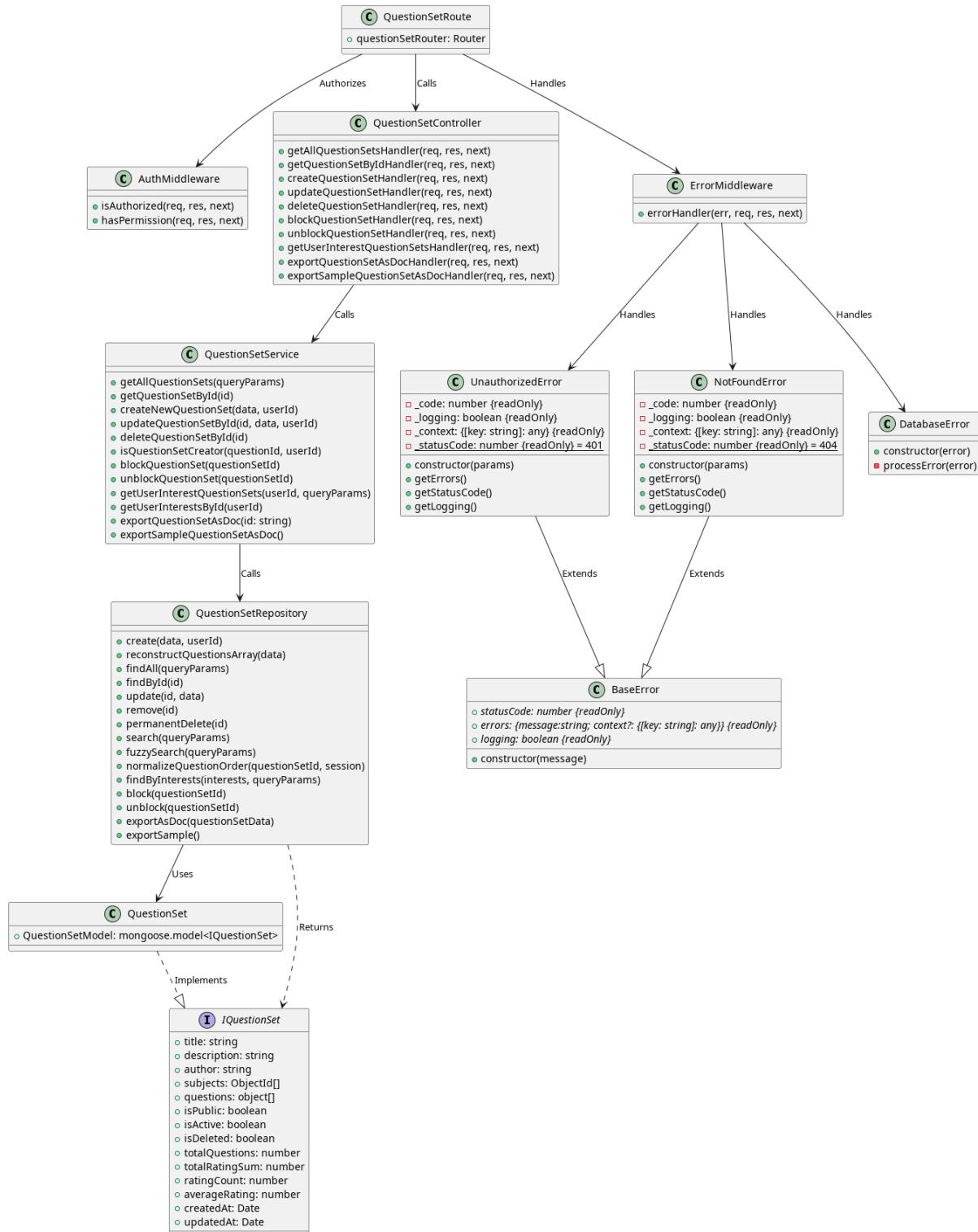


Figure 3.3.7.1: Export question set⁷⁵

⁷⁵ [export question set.png](#)

3.3.7.2 Sequence diagram

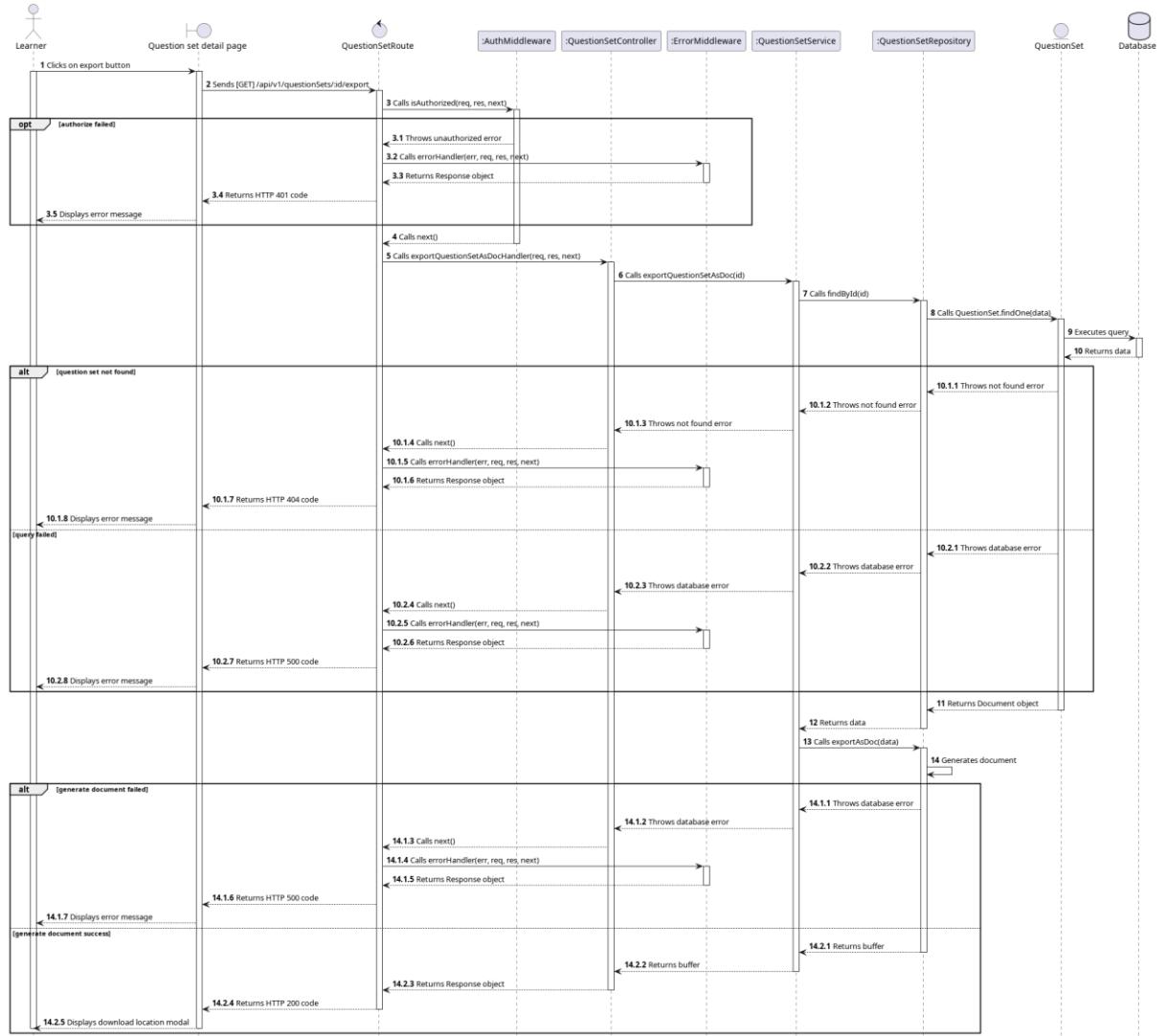


Figure 3.3.7.2: Export question set⁷⁶

⁷⁶ [export question set.png](#)

3.3.8 Rate question sets

3.3.8.1 Class diagram

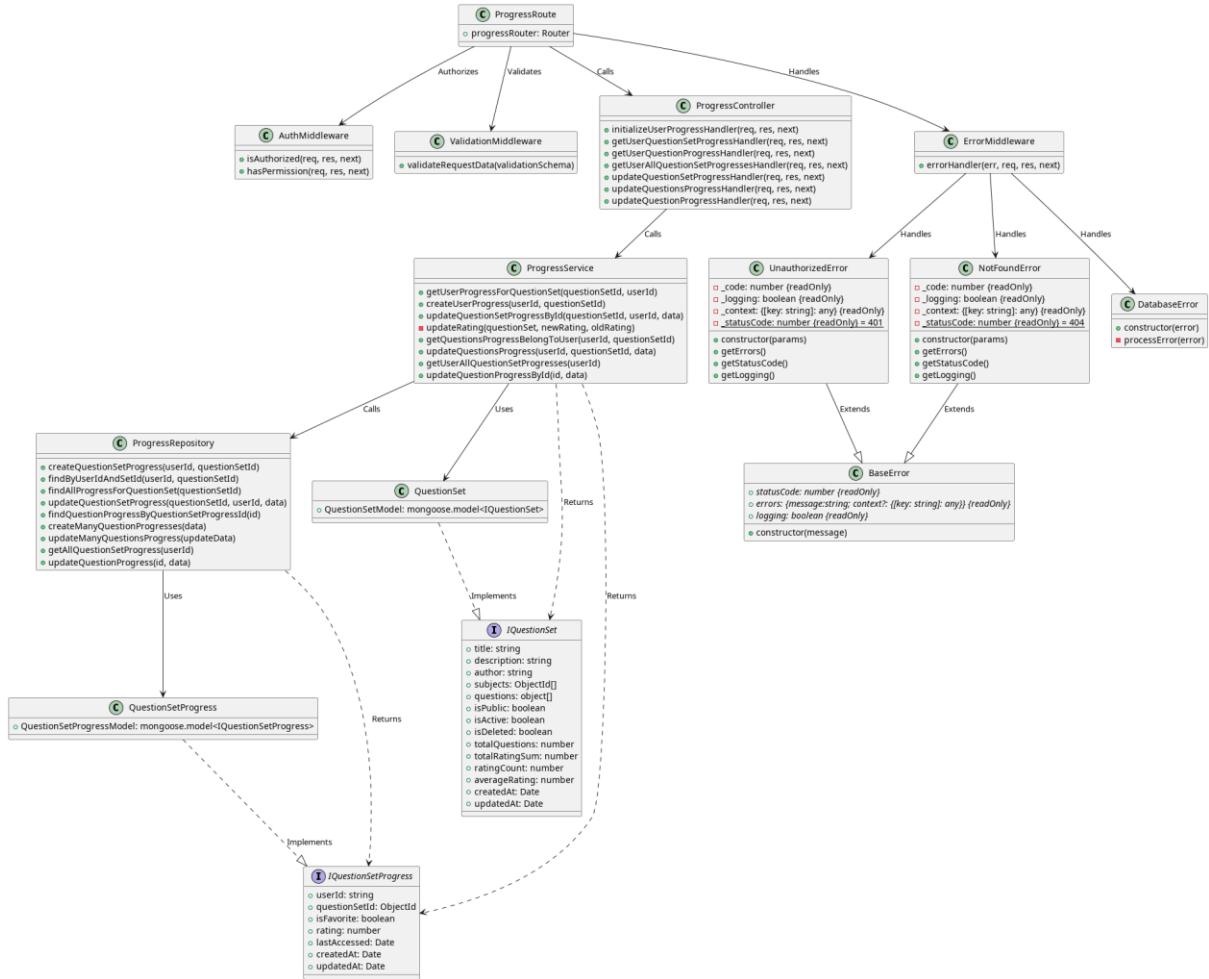


Figure 3.3.8.1: Rate question set⁷⁷

⁷⁷ [rate question set.png](#)

3.3.8.2 Sequence diagram

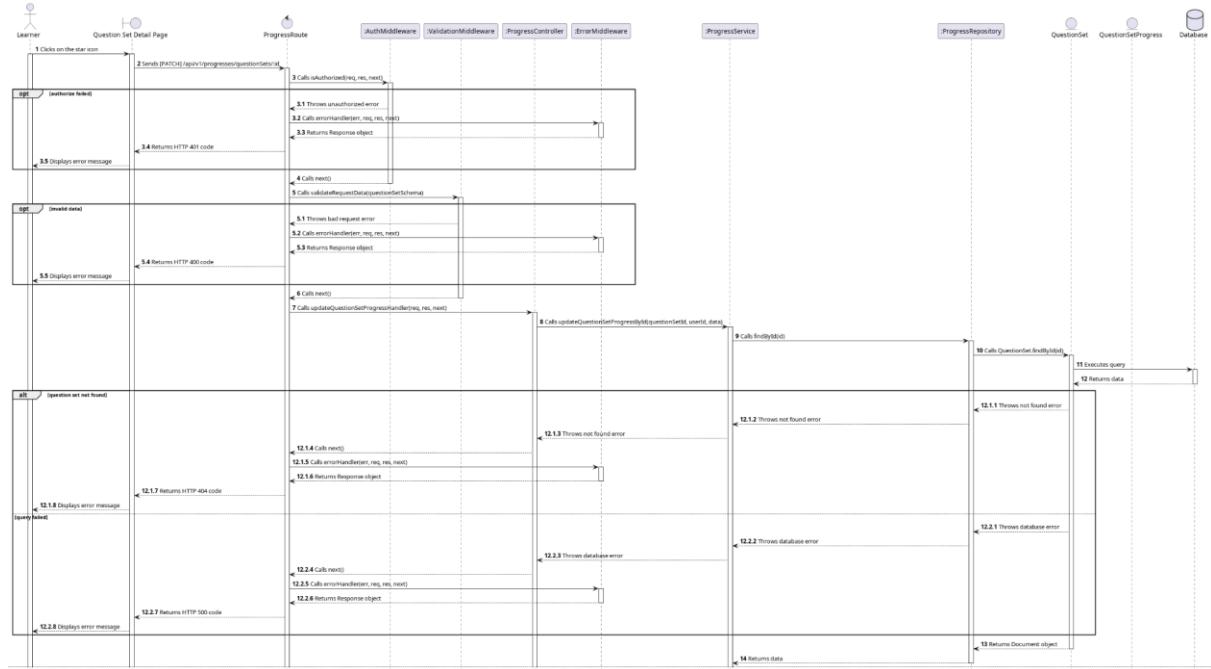


Figure 3.3.8.2.a: Rate question set - 1⁷⁸

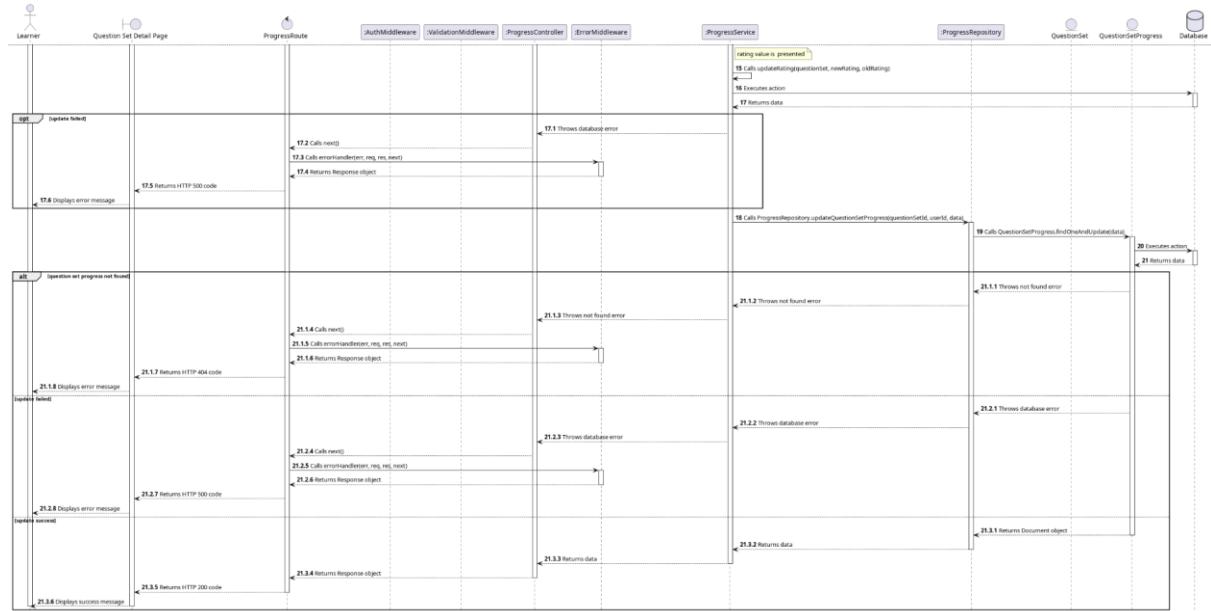


Figure 3.3.8.2.b: Rate question set - 2

⁷⁸ rate question set.png

3.3.9 Toggle favourite

3.3.9.1 Class diagram

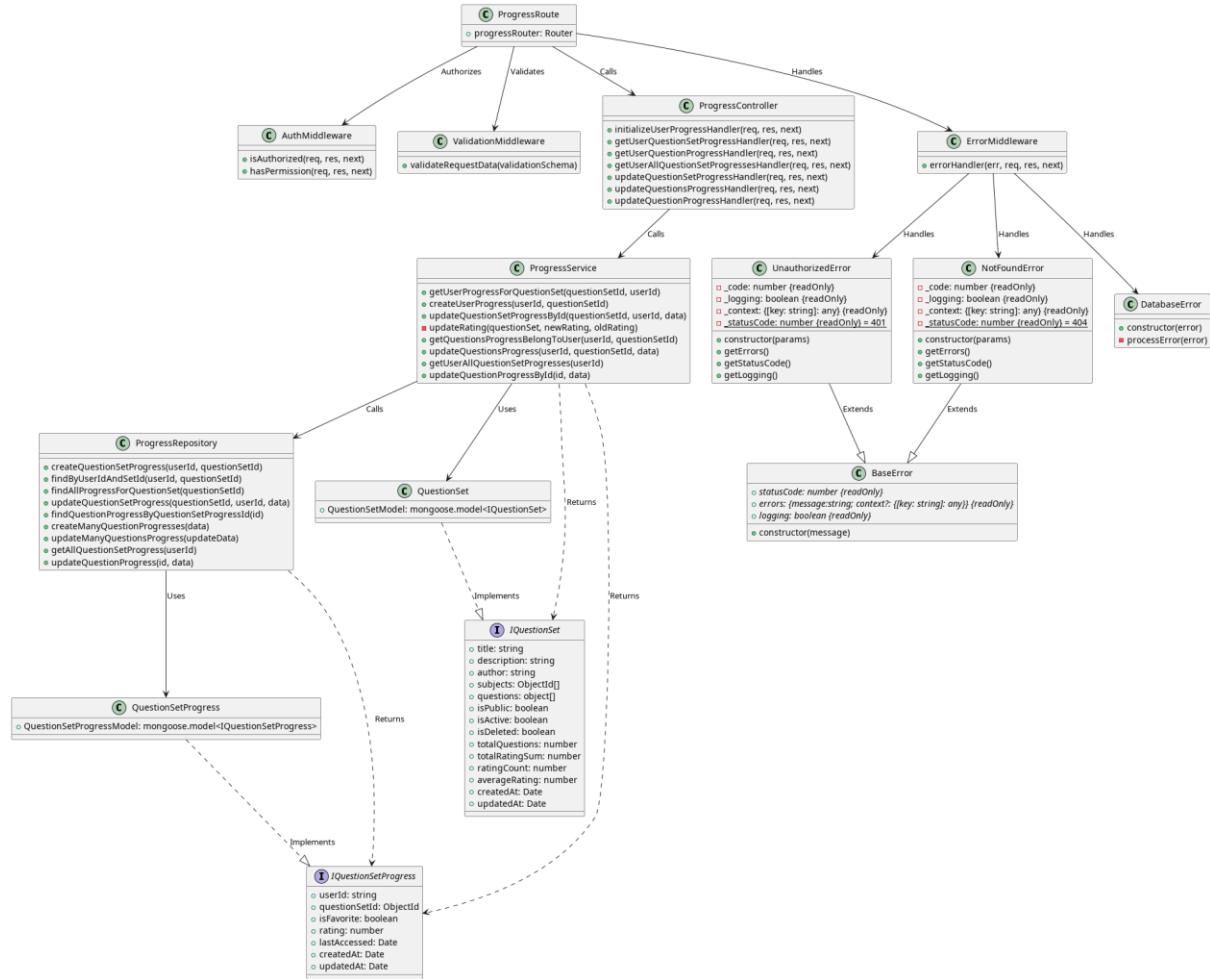


Figure 3.3.9.1: Toggle favourite⁷⁹

⁷⁹ [toggle favorite.png](#)

3.3.9.2 Sequence diagram

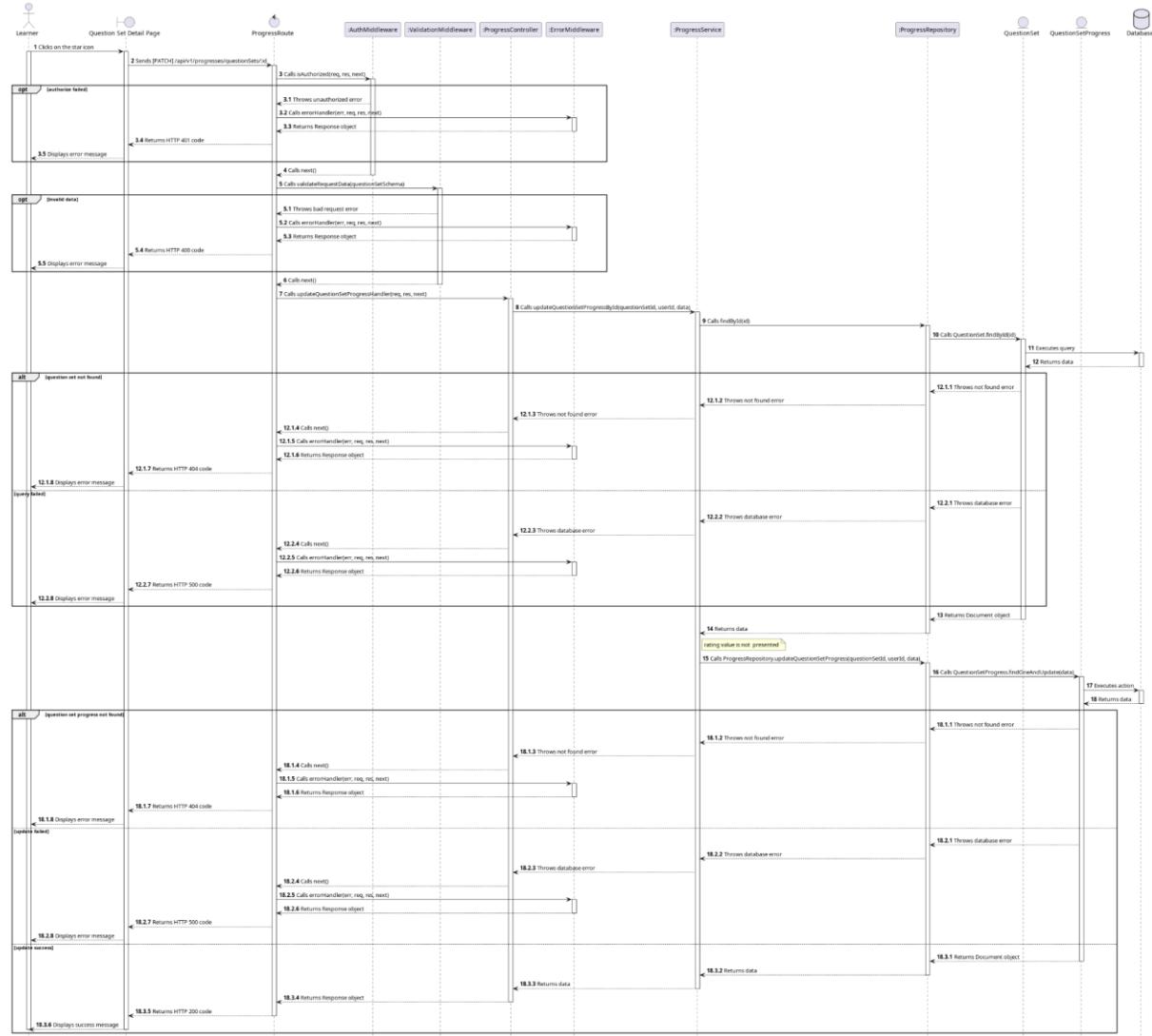


Figure 3.3.9.2: Toggle favourite⁸⁰

⁸⁰ toggle favorite.png

3.3.10 Update question set accessibilities

3.3.10.1 Class diagram

3.3.10.1.1 Block question set

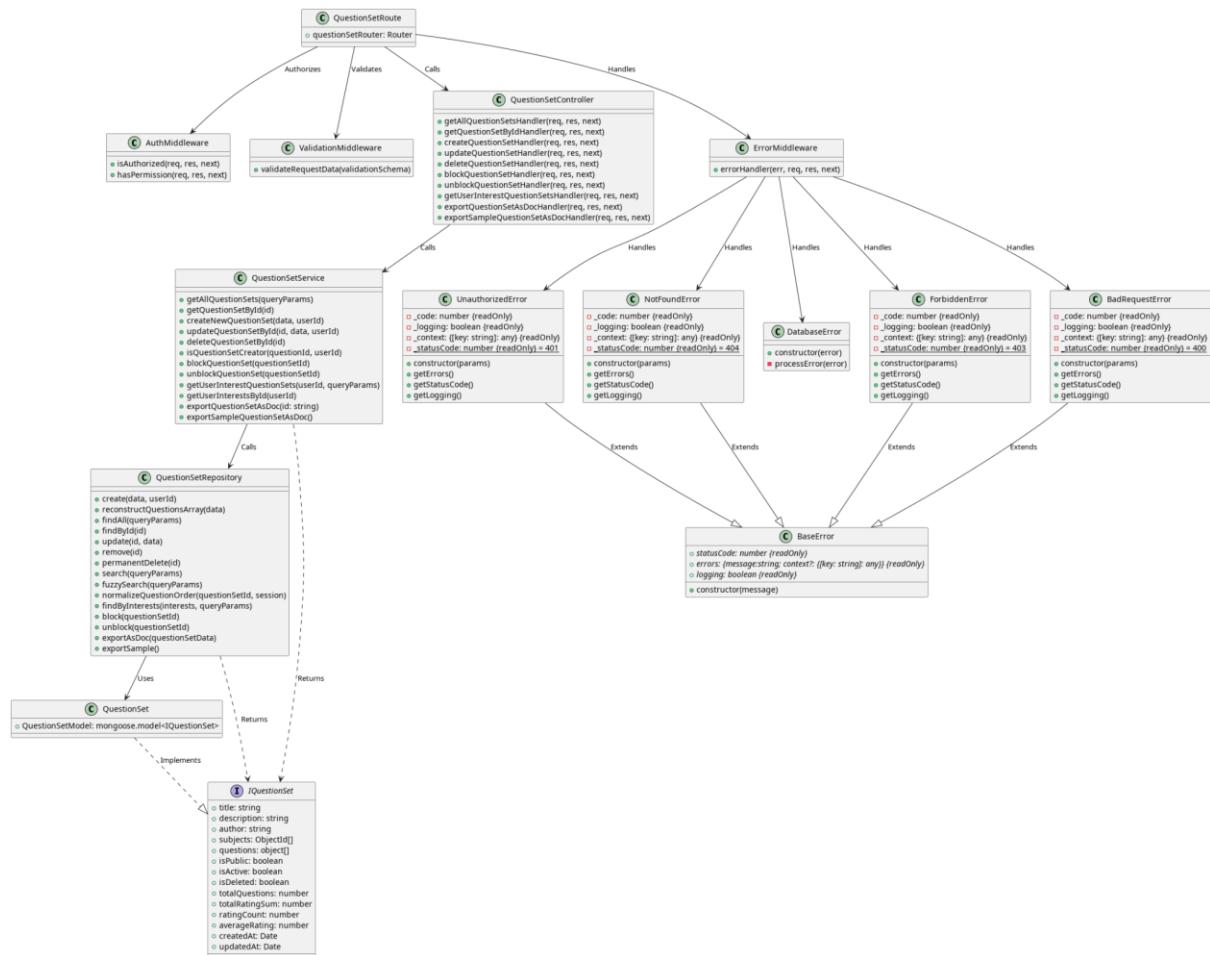


Figure 3.3.10.1.1: Block question set⁸¹

⁸¹ [block question set.png](#)

3.3.10.1.2 Unblock question set

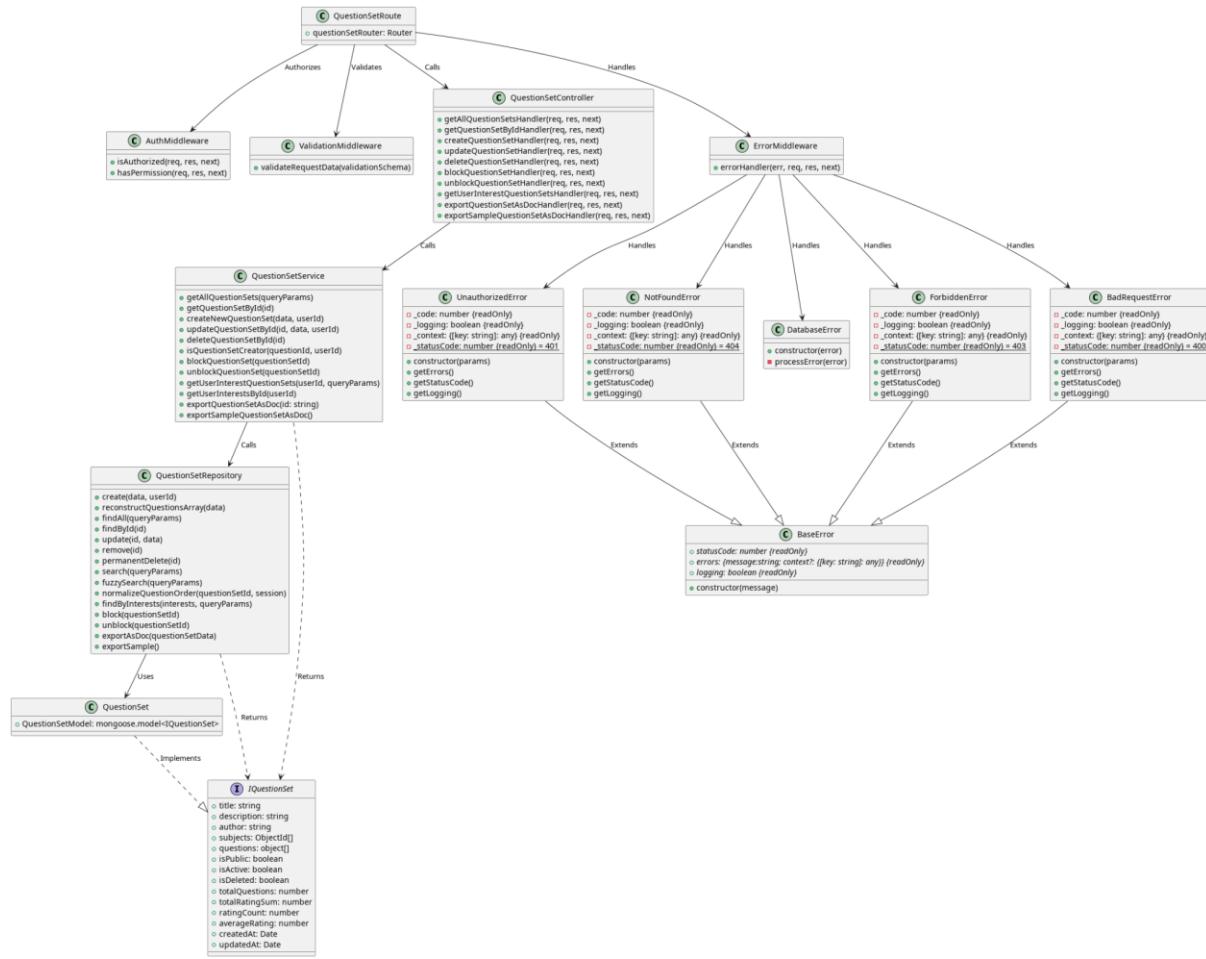


Figure 3.3.10.1.2: Unblock question set⁸²

⁸² [unblock question set.png](#)

3.3.10.2 Sequence diagram

3.3.10.2.1 Block question set

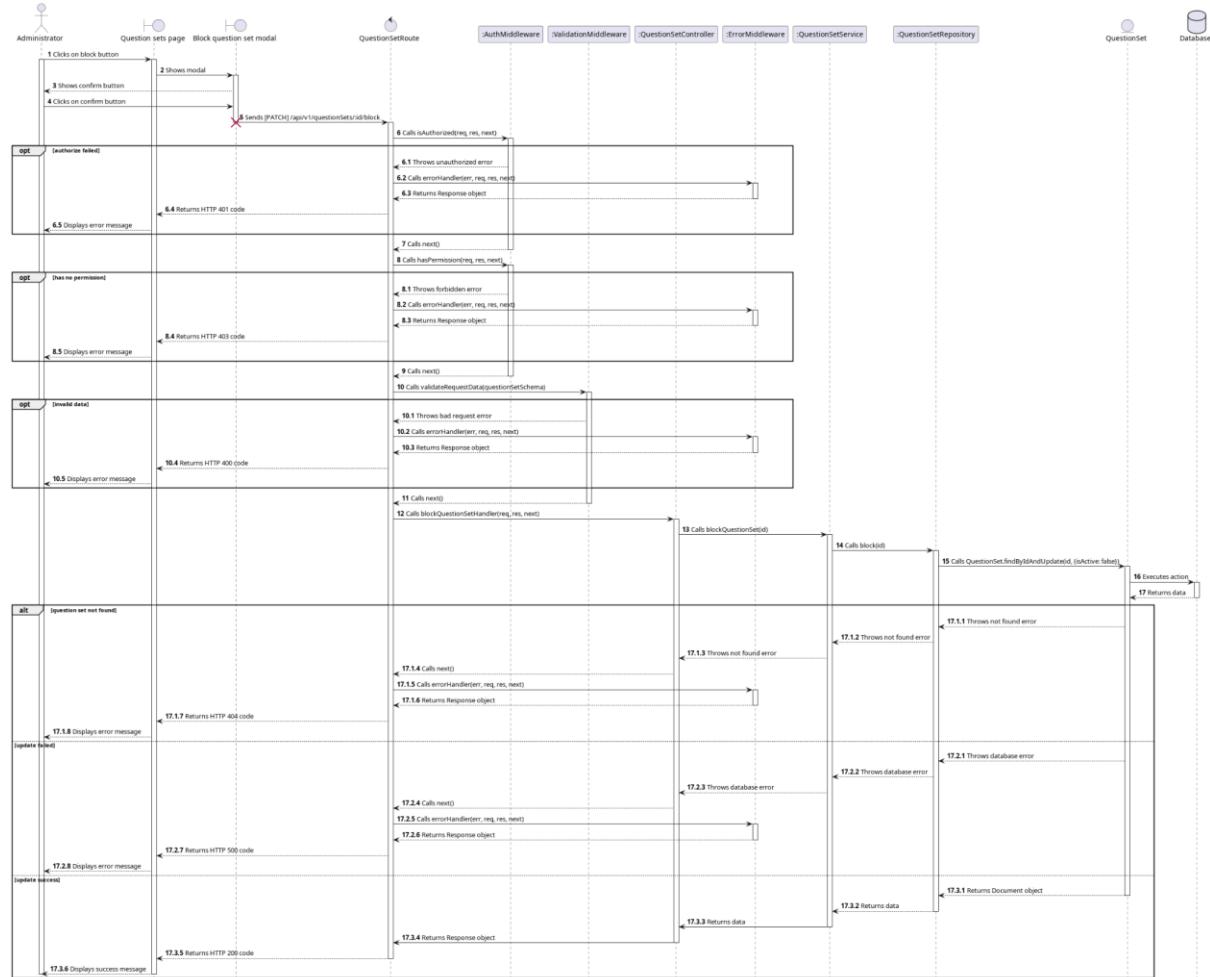


Figure 3.3.10.2.1: Block question set⁸³

⁸³ [block question set.png](#)

3.3.10.2.2 Unblock question set

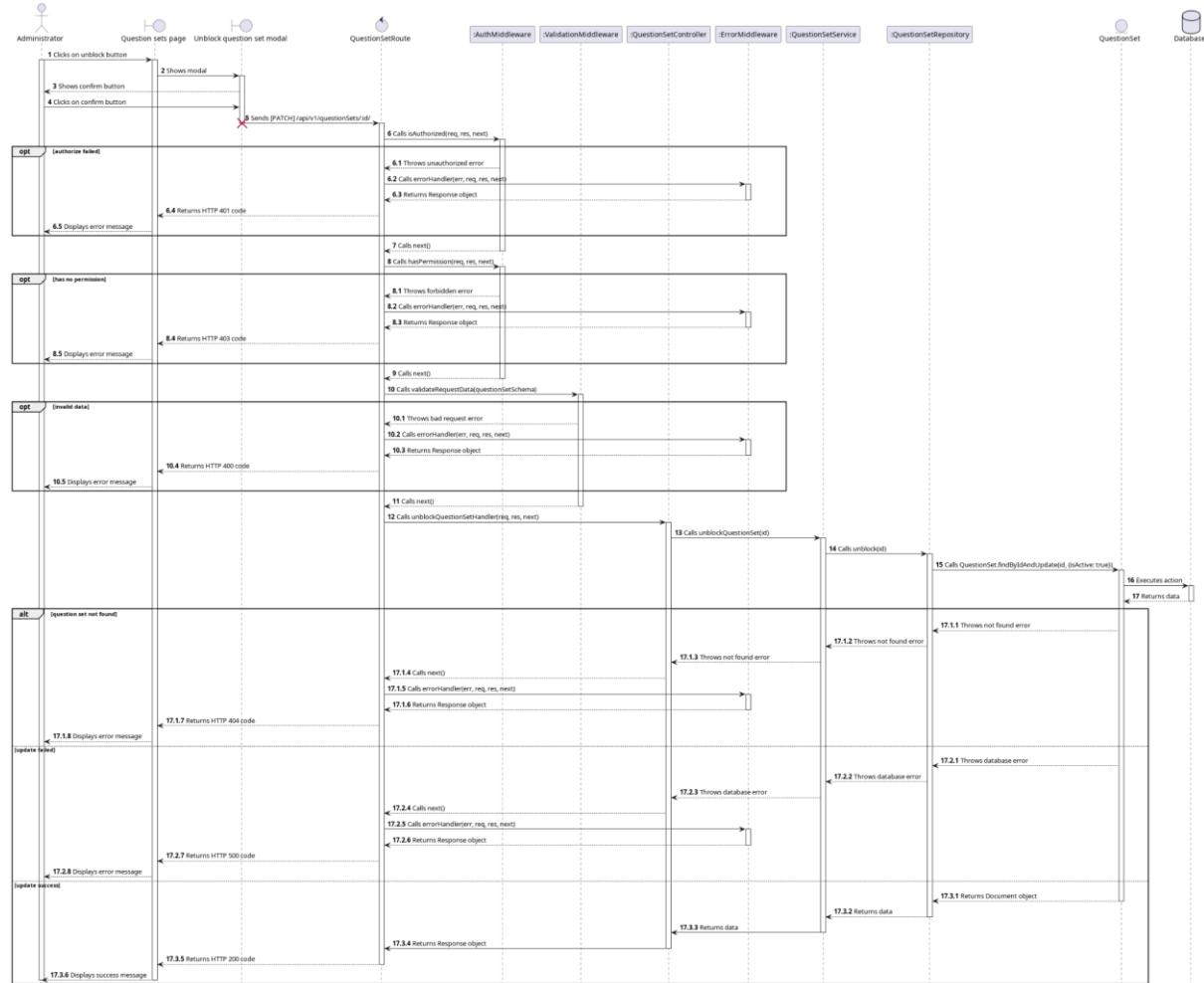


Figure 3.3.10.2.2: Unblock question set⁸⁴

84 unblock question set.png

3.4 Group Management

3.4.1 View groups

3.4.1.1 Class diagram



Figure 3.4.1.1 View groups⁸⁵

⁸⁵ [view groups.png](#)

3.4.1.2 Sequence diagram

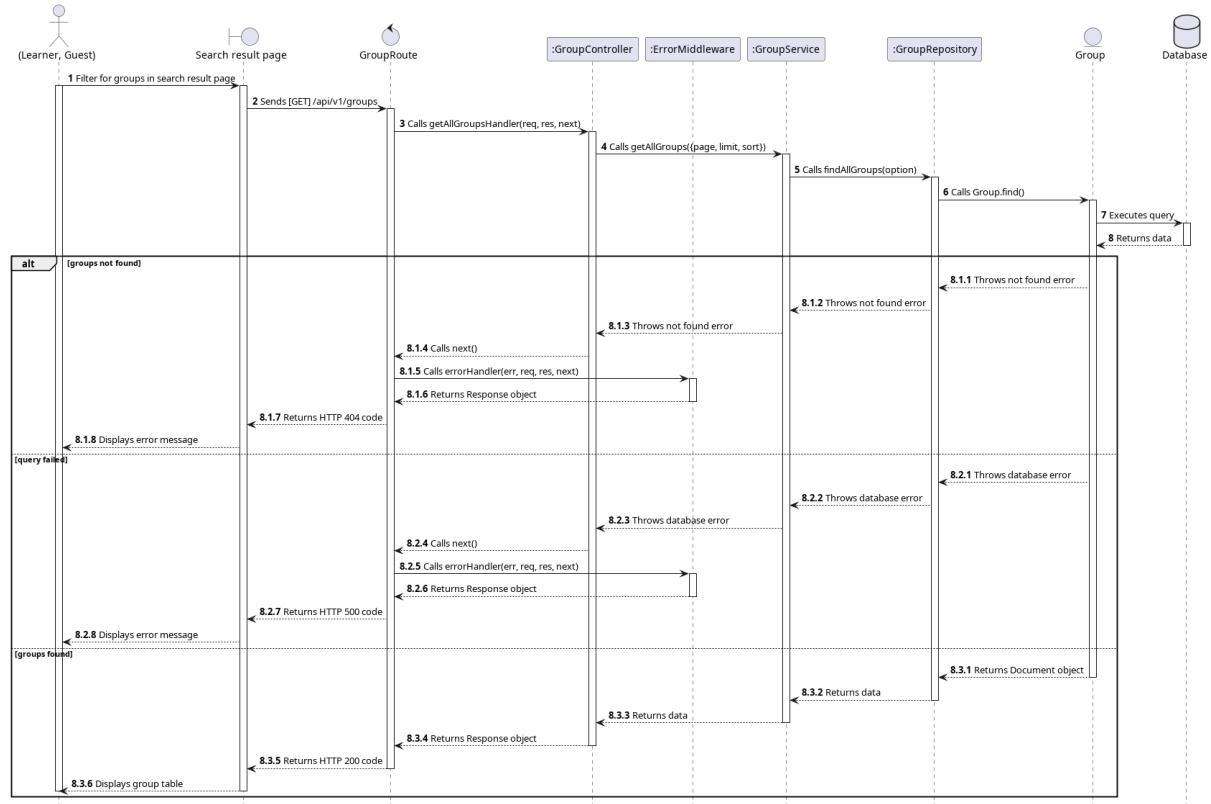


Figure 3.4.1.2.a Learner and guests view groups⁸⁶

⁸⁶ [view groups.png](#)

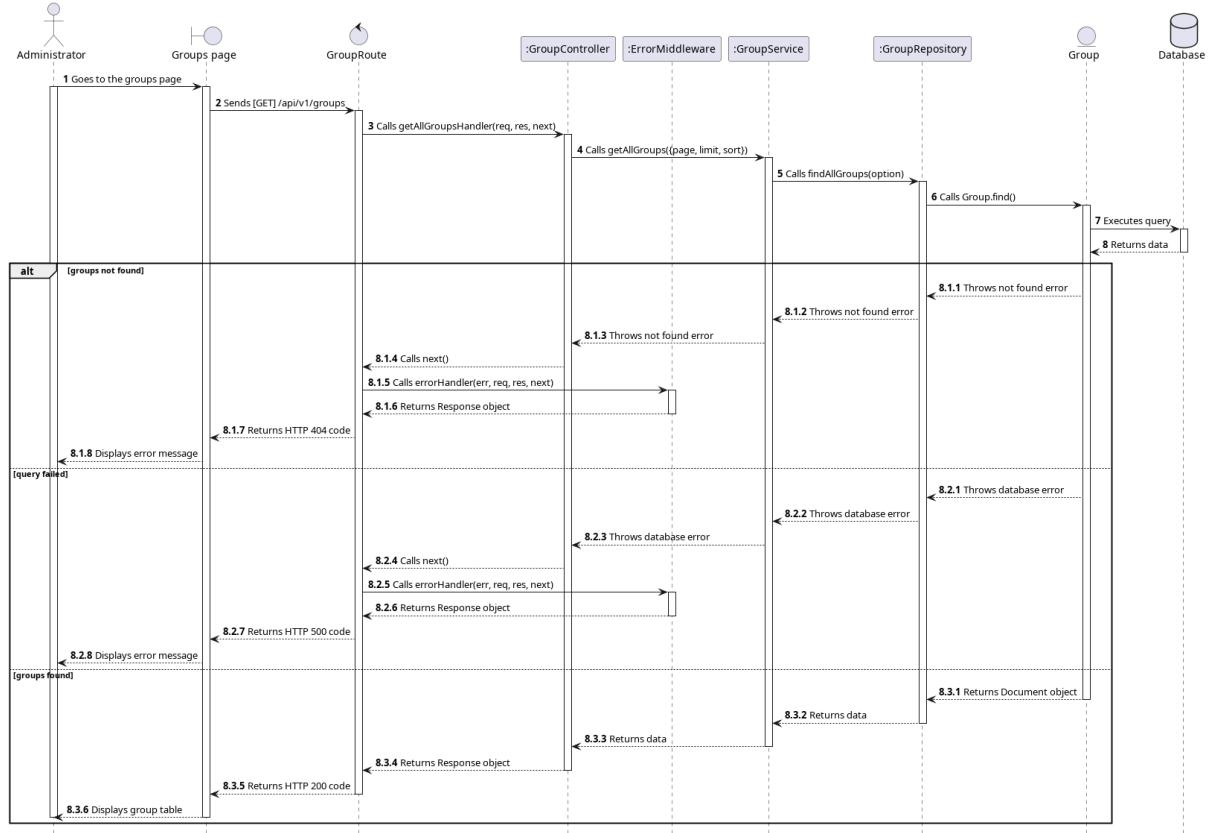


Figure 3.4.1.2.b Administrator view groups⁸⁷

⁸⁷ [view groups.png](#)

3.4.2 View group detail

3.4.2.1 Class diagram



Figure 3.4.2.1 View group detail⁸⁸

88 view group detail.png

3.4.2.2 Sequence diagram

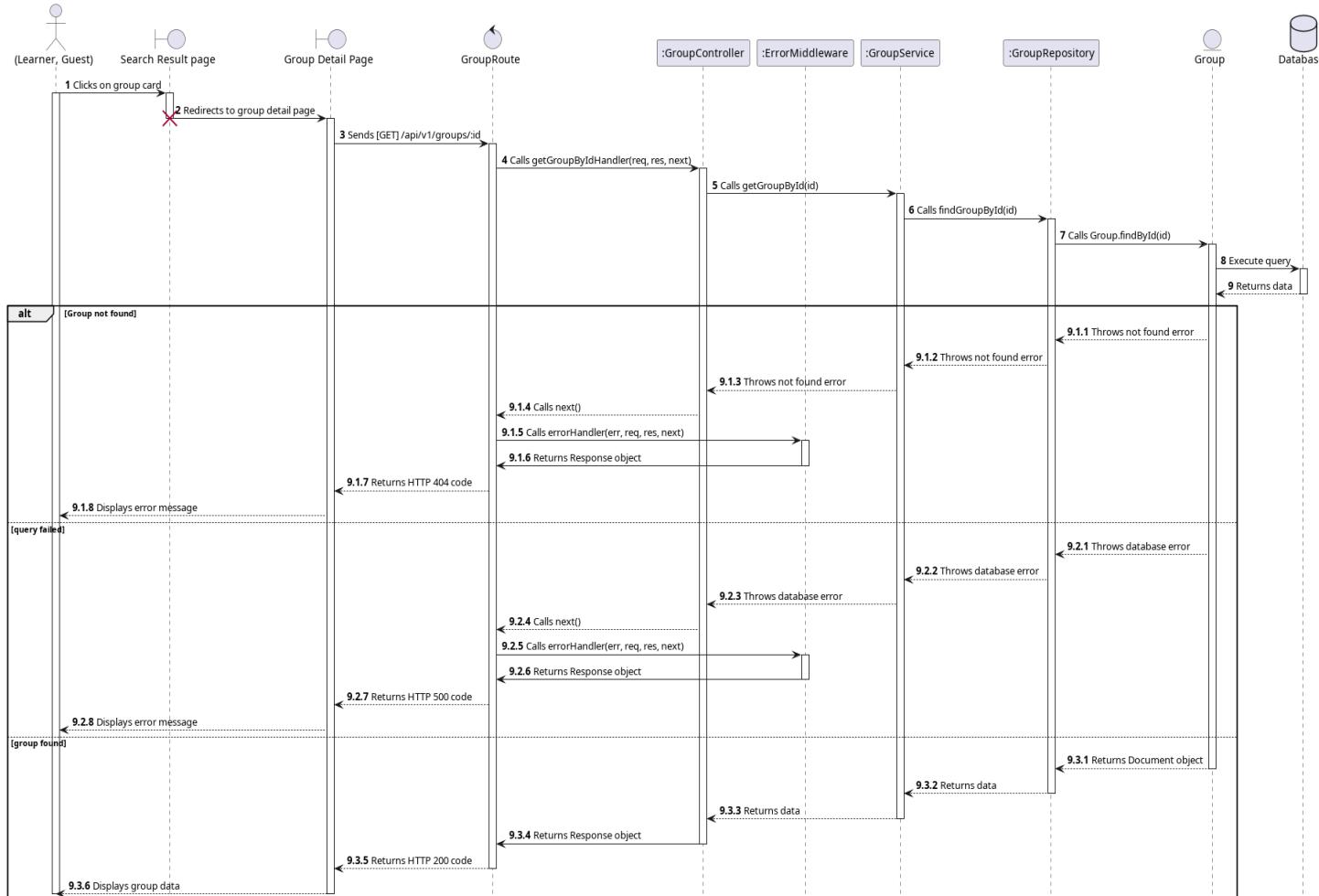


Figure 3.4.2.2.a Learner and Guest view group detail⁸⁹

⁸⁹ [view group detail.png](#)

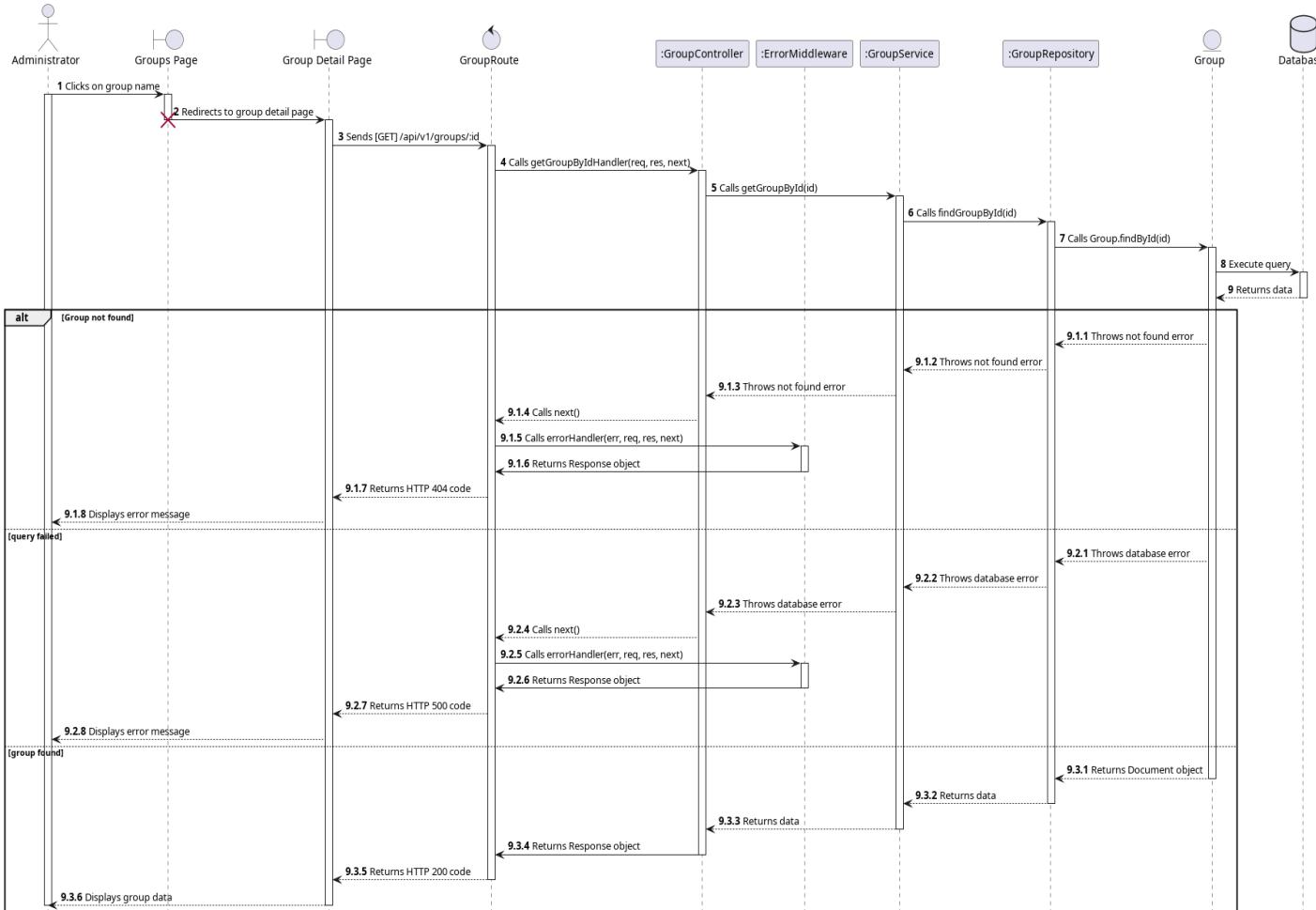


Figure 3.4.2.2.b Administrator view group detail⁹⁰

⁹⁰ [view group detail.png](#)

3.4.3 Create groups

3.4.3.1 Class diagram

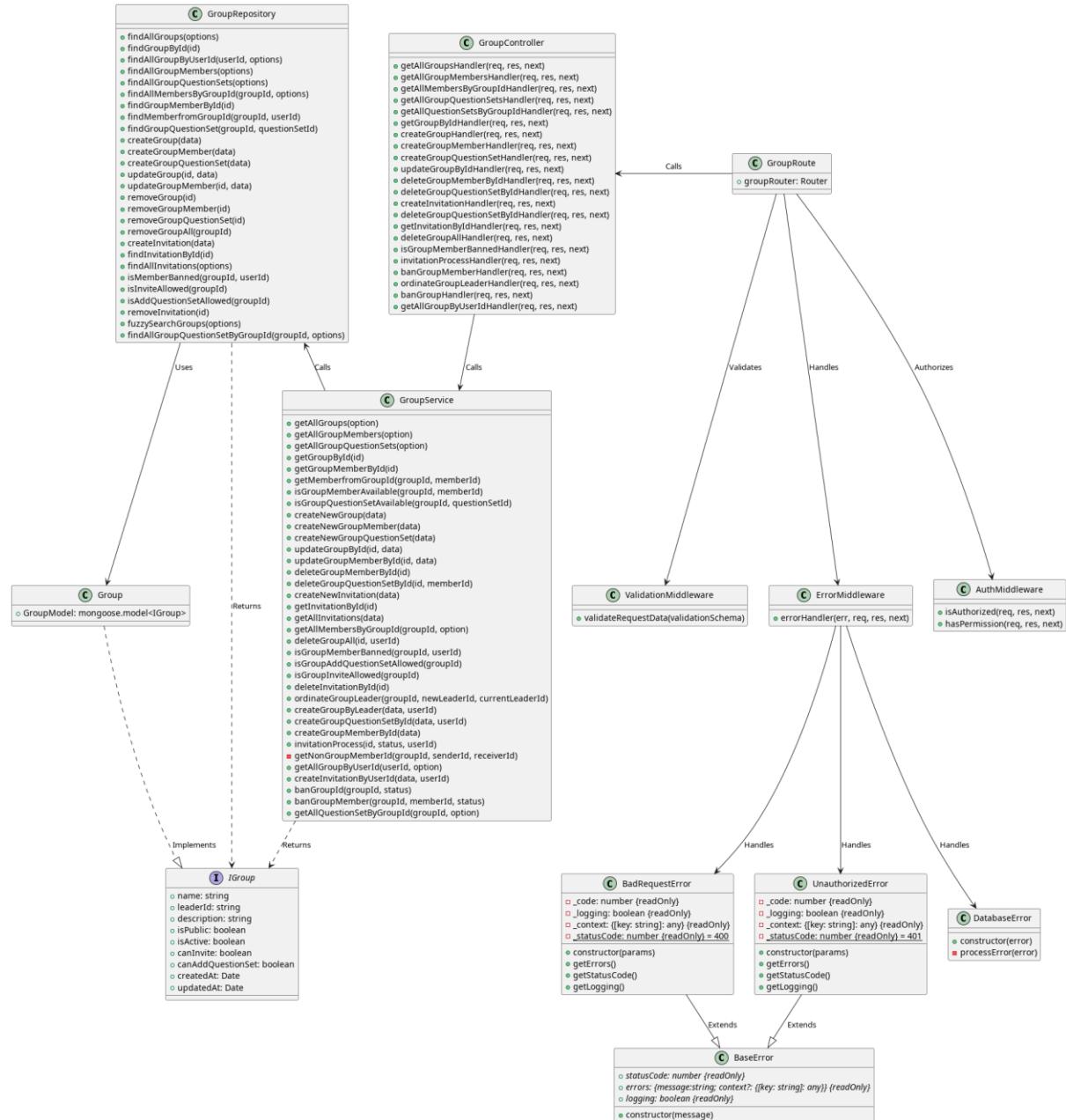


Figure 3.4.3.1 Create groups⁹¹

91 [create group.png](#)

3.4.3.2 Sequence diagram

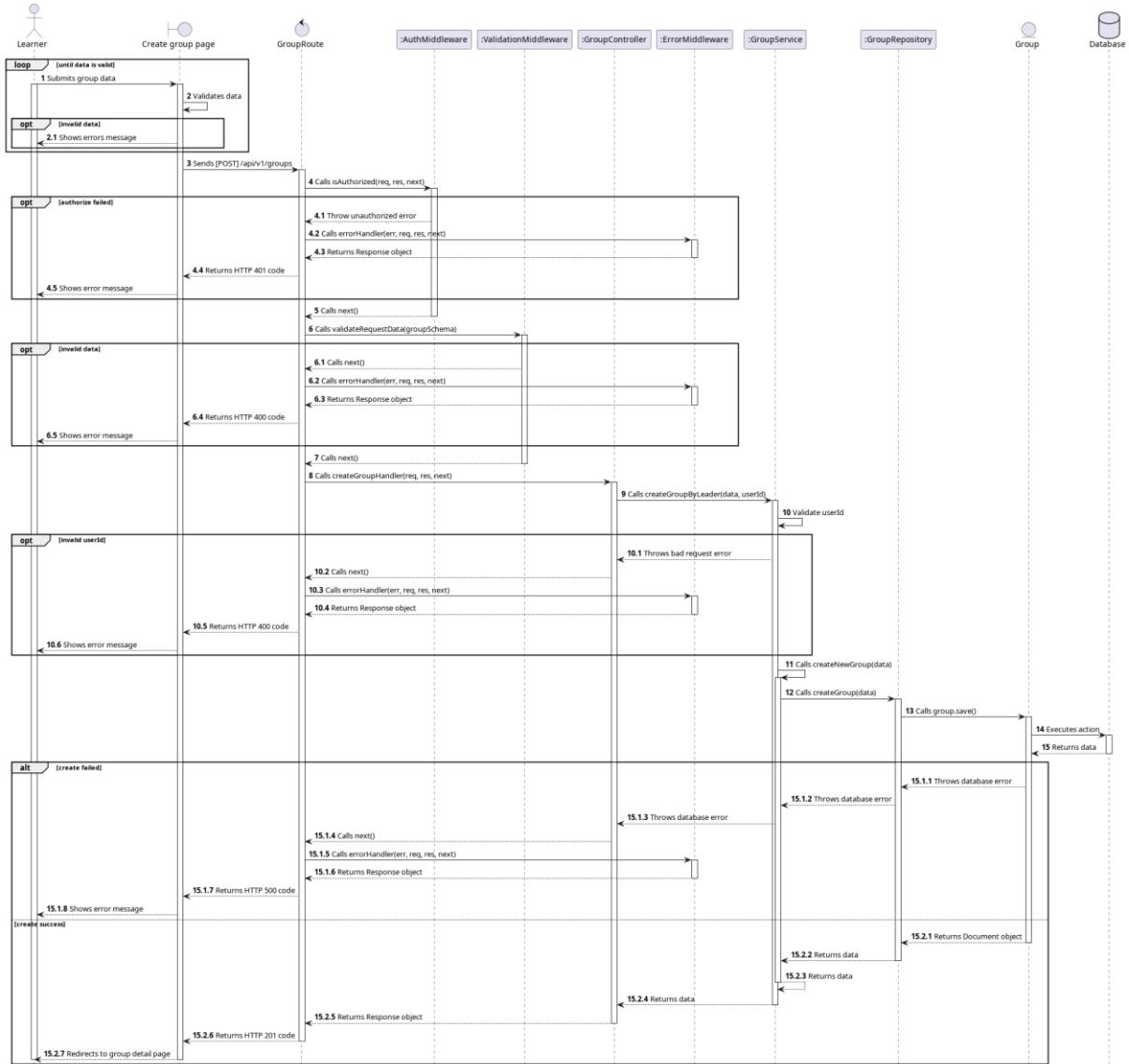


Figure 3.4.3.2 Create groups⁹²

⁹² [create group.png](#)

3.4.4 Update groups

3.4.4.1 Class diagram

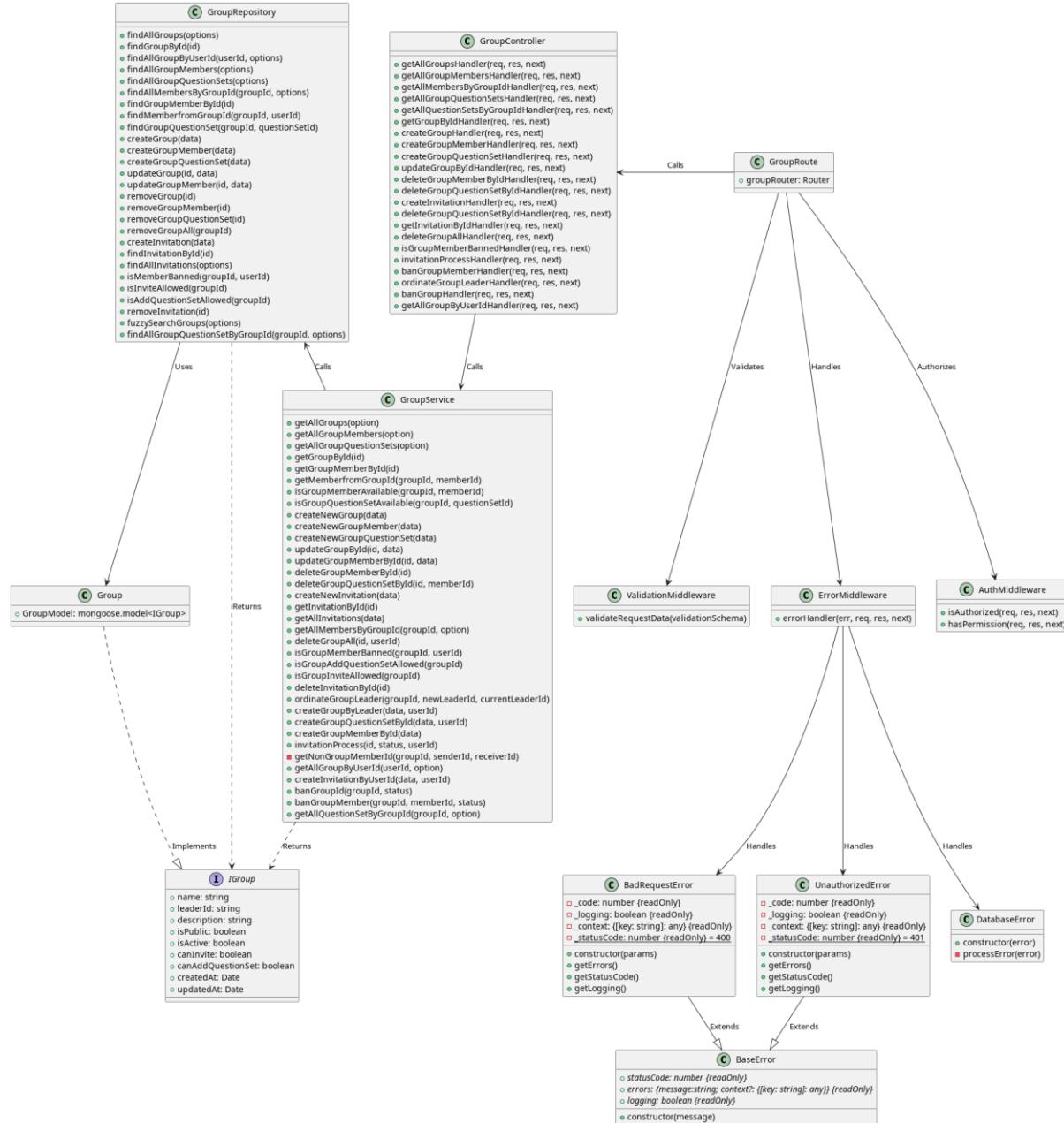


Figure 3.4.4.1 Update groups⁹³

⁹³ [update group.png](#)

3.4.4.2 Sequence diagram

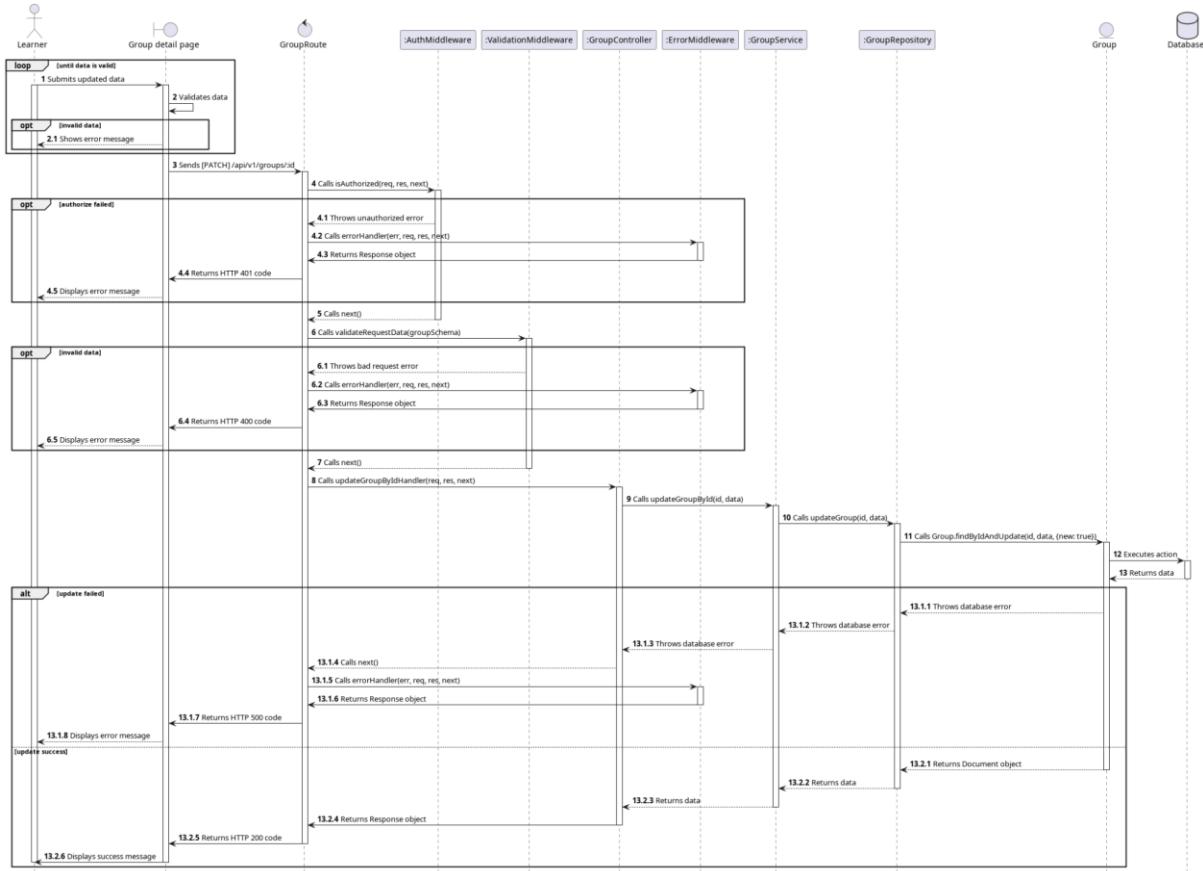


Figure 3.4.4.2 Update groups⁹⁴

⁹⁴ [update group.png](#)

3.4.5 Delete groups

3.4.5.1 Class diagram

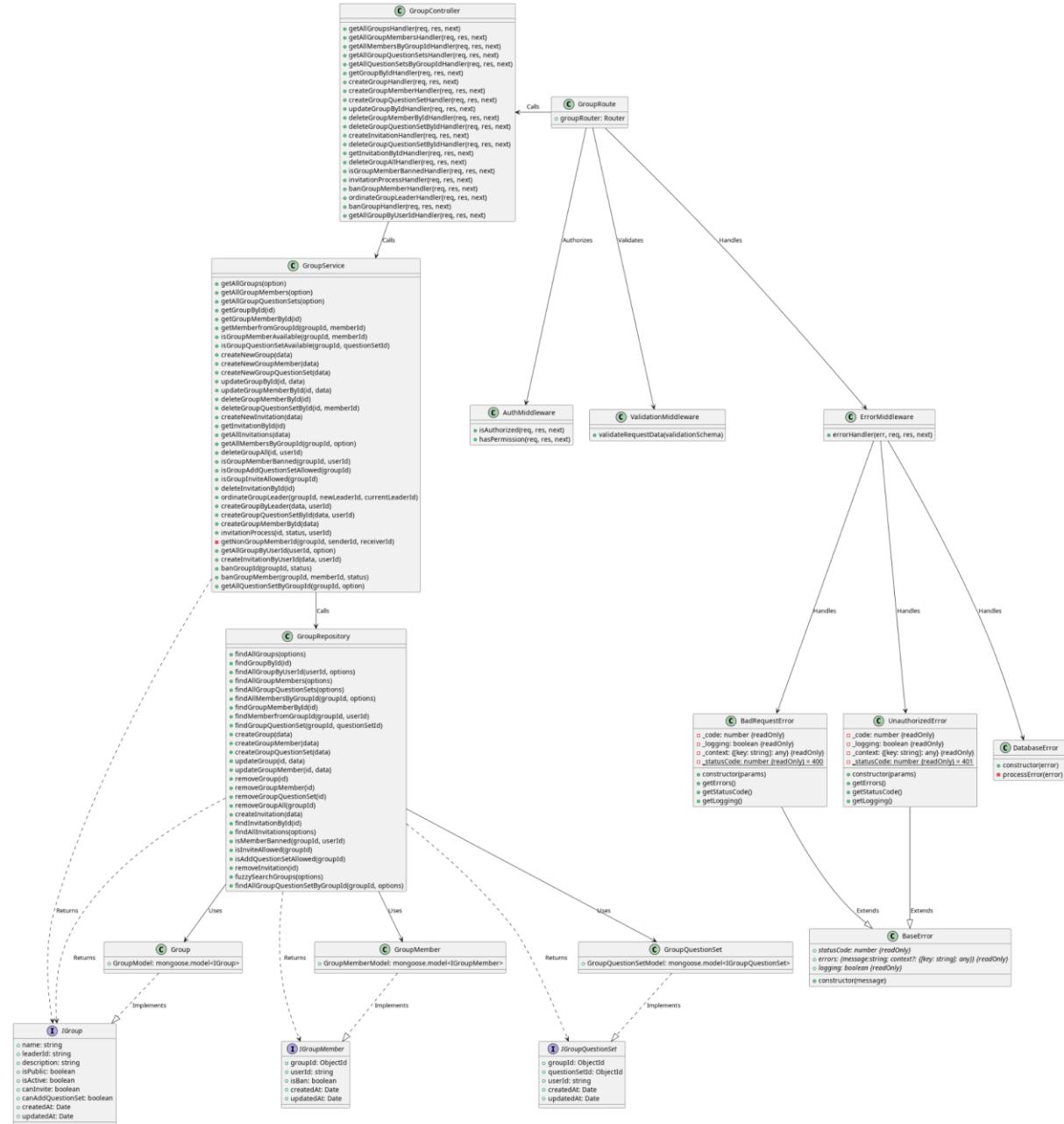


Figure 3.4.5.1 Delete groups⁹⁵

⁹⁵ [delete group.png](#)

3.4.5.2 Sequence diagram

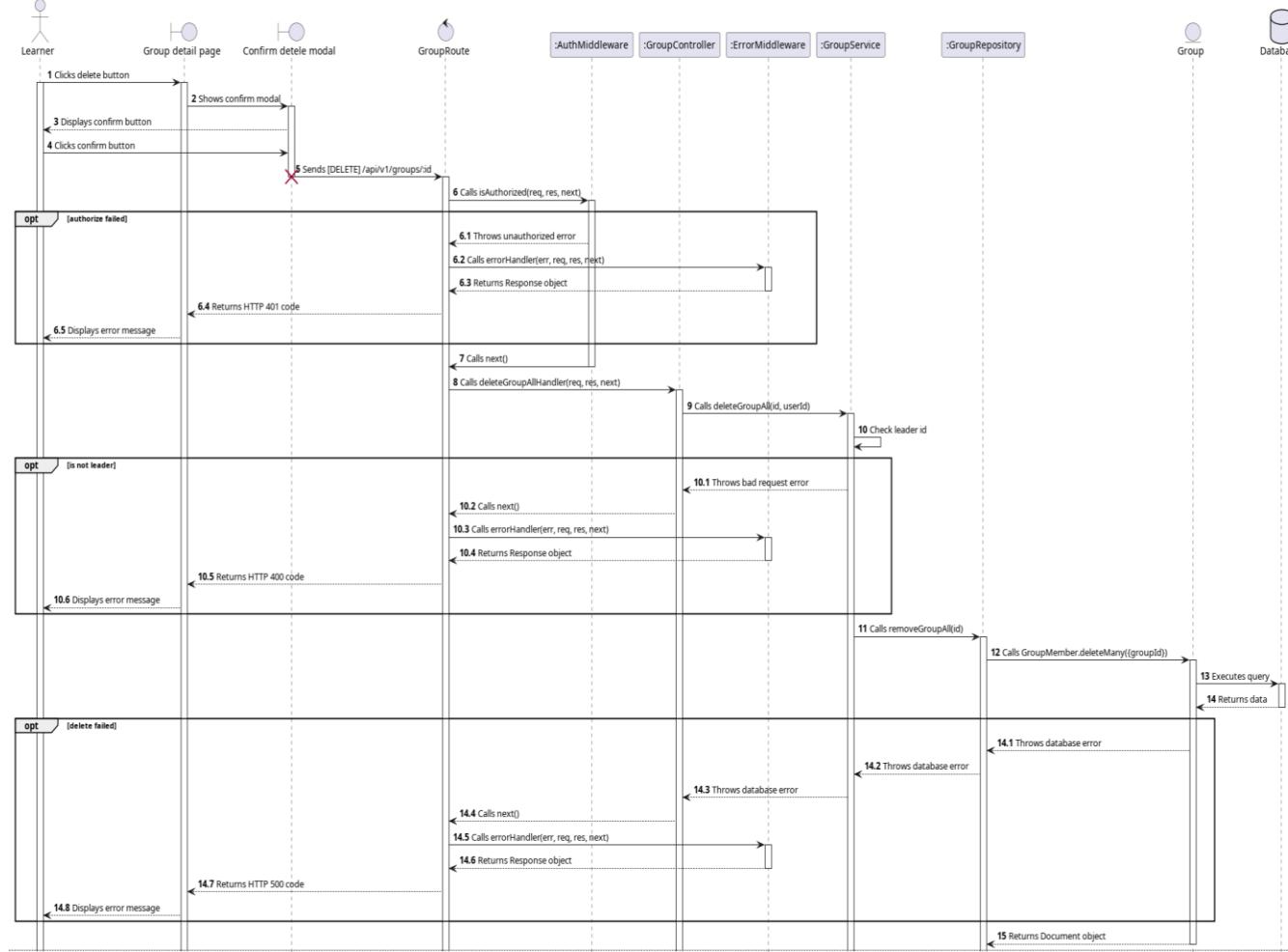


Figure 3.4.5.2.a Delete groups - 1⁹⁶

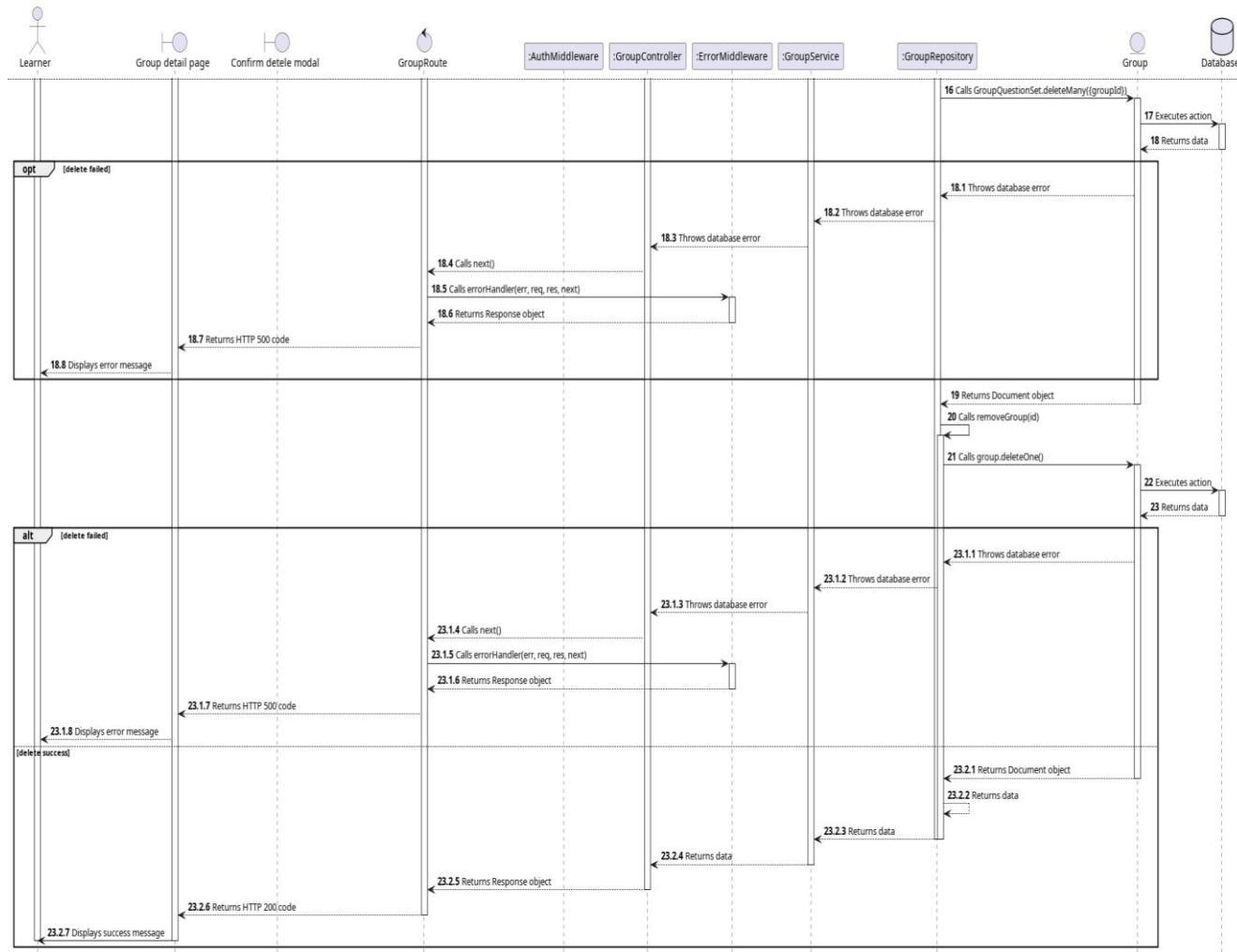


Figure 3.4.5.2.b Delete groups - 2

3.4.6 Send invitations

3.4.6.1 Class diagram

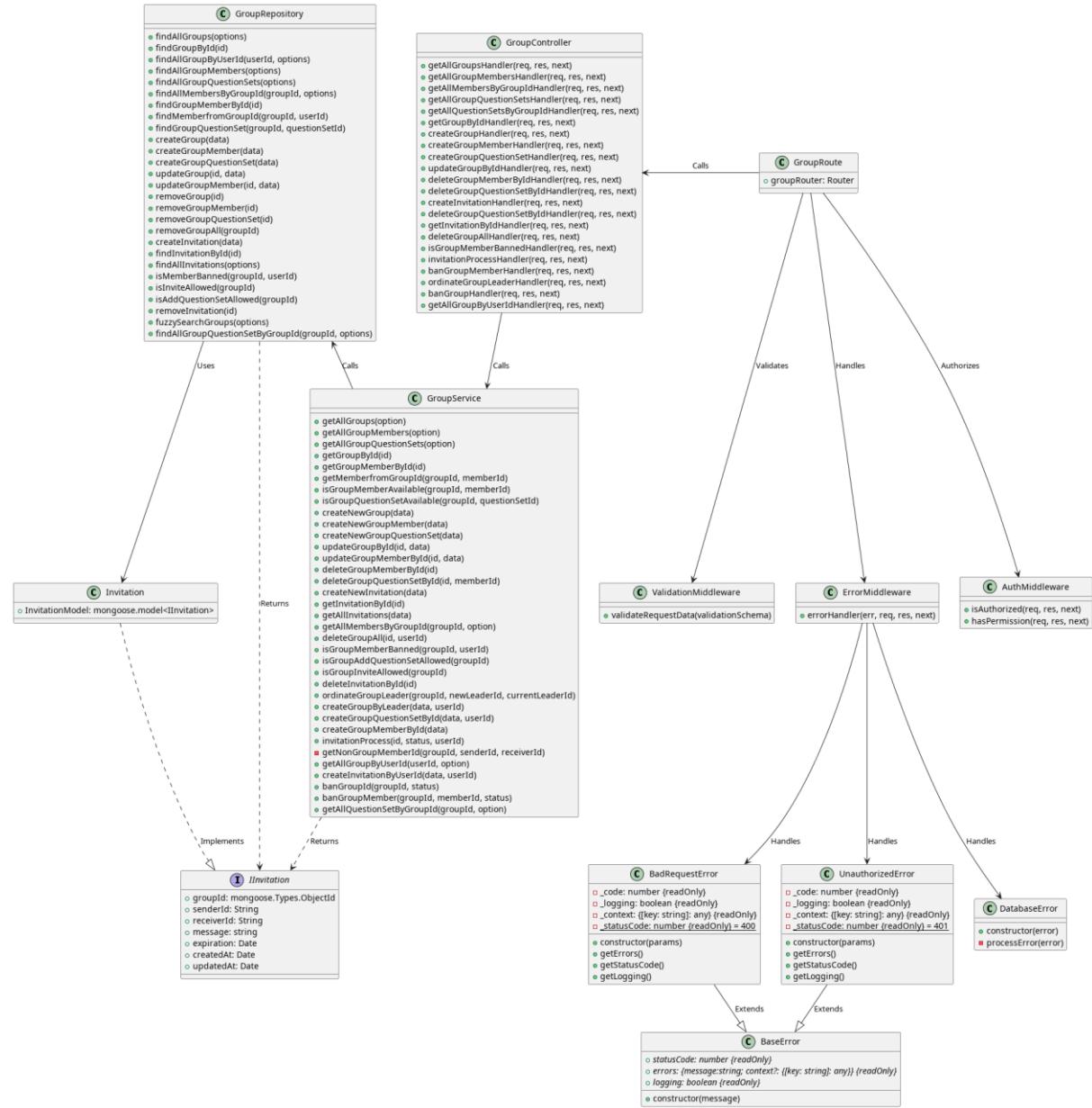


Figure 3.4.6.1 Send invitations⁹⁷

97 [send invitation.png](#)

3.4.6.2 Sequence diagram

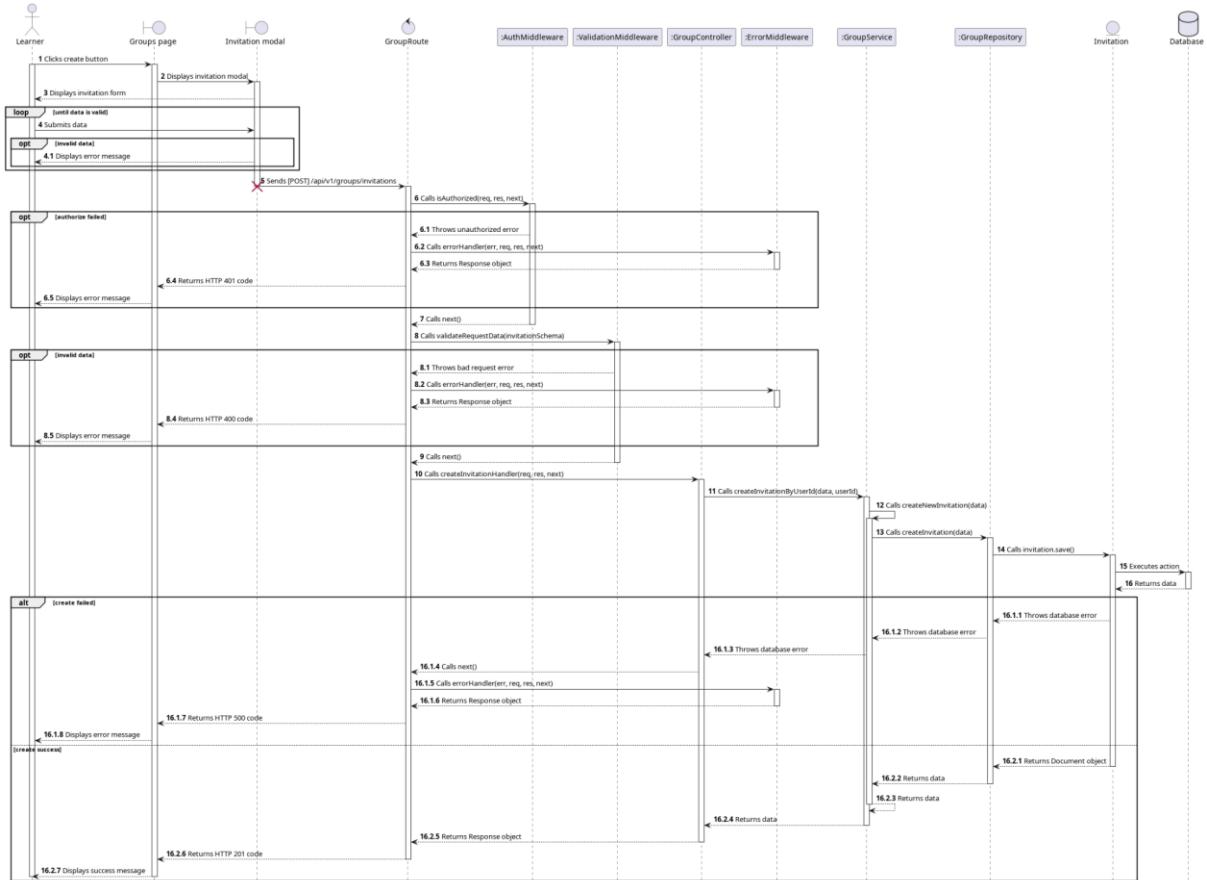


Figure 3.4.6.2 Send invitations⁹⁸

⁹⁸ [send invitations.png](#)

3.4.7 Join groups

3.4.7.1 Class diagram

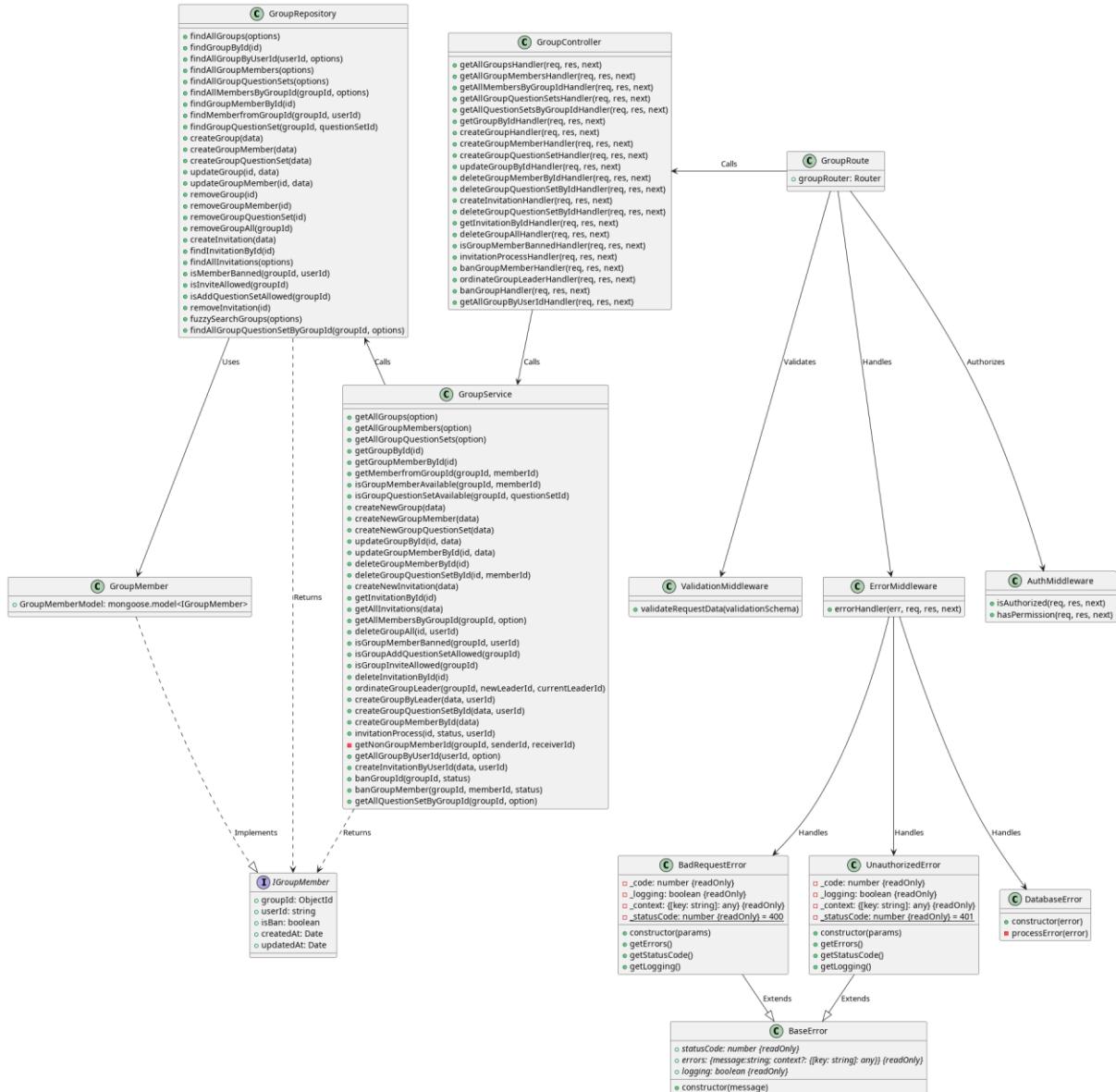


Figure 3.4.7.1 Join groups⁹⁹

⁹⁹ [join group.png](#)

3.4.7.2 Sequence diagram

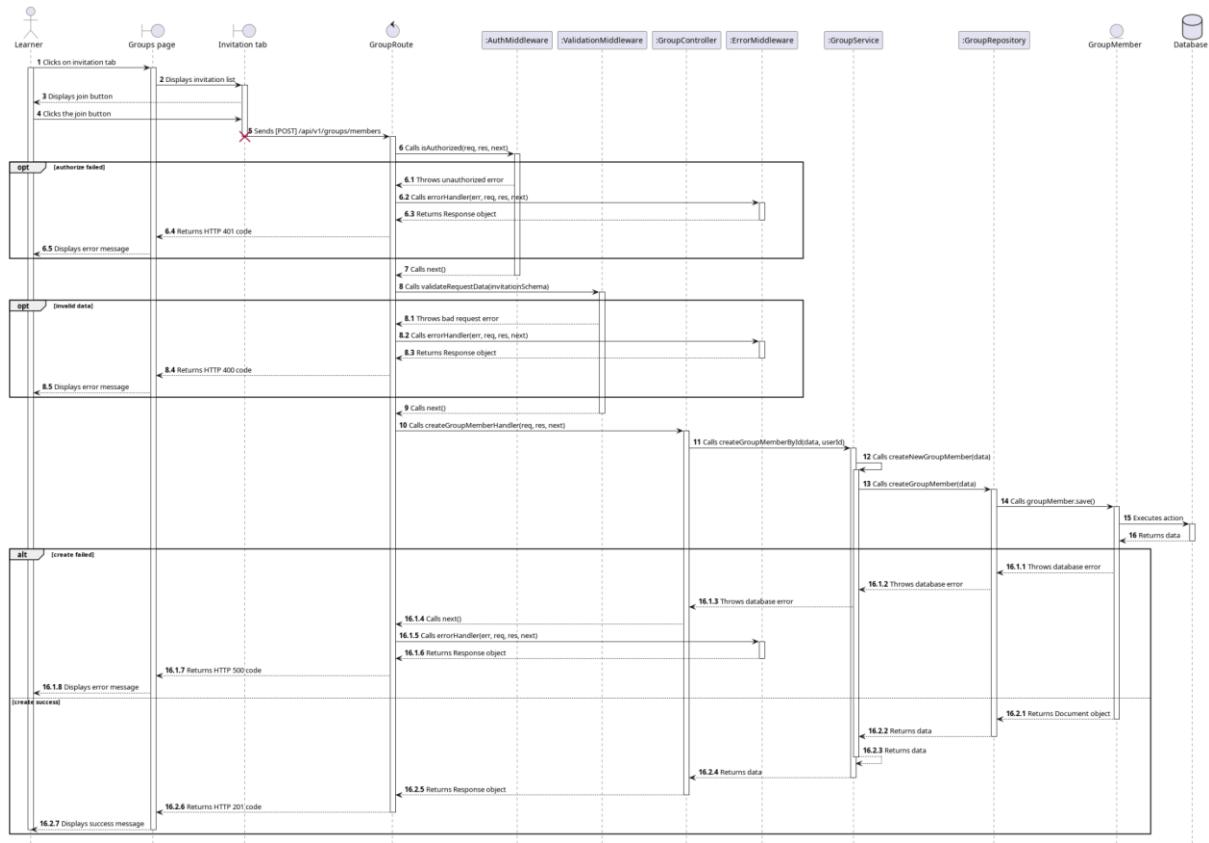


Figure 3.4.7.2 Join groups¹⁰⁰

¹⁰⁰ [join group.png](#)

3.4.8 Leave groups

3.4.8.1 Class diagram

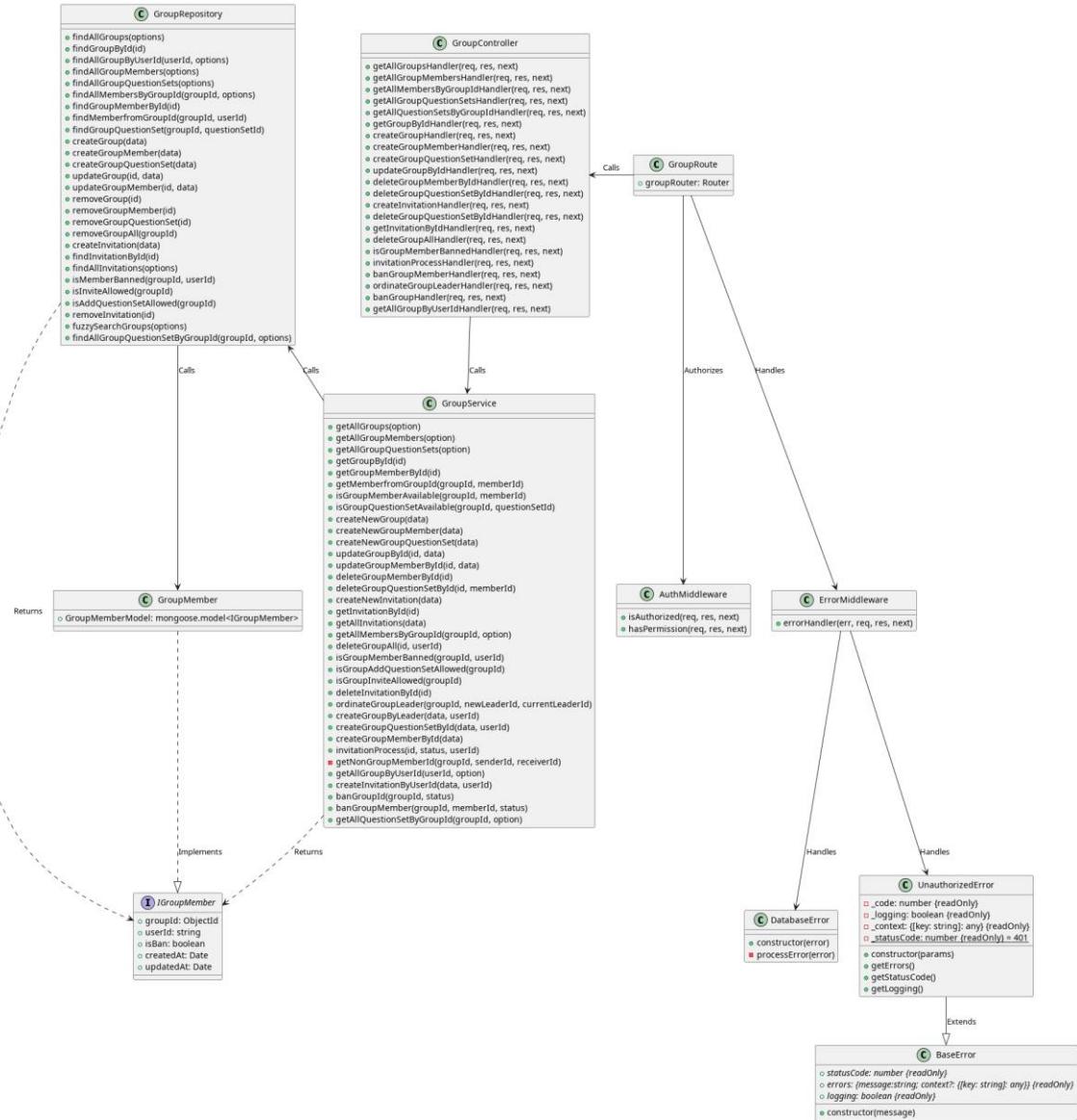


Figure 3.4.8.1 Leave groups¹⁰¹

101 leave group.png

3.4.8.2 Sequence diagram

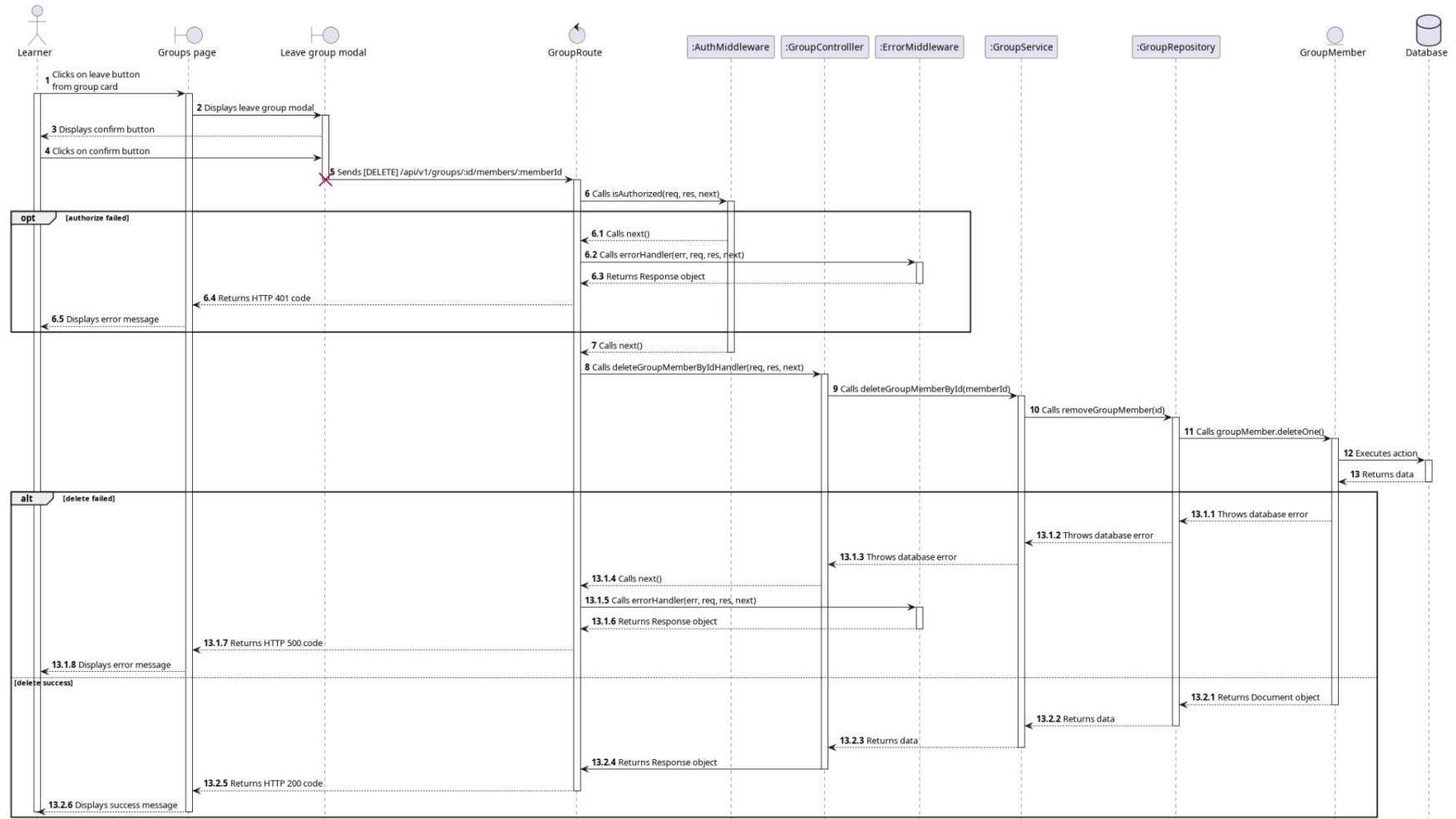


Figure 3.4.8.2 Leave groups¹⁰²

102 [leave group.png](#)

3.4.9 Update groups accessibilities

3.4.9.1 Class diagram

3.4.9.1.1 Block group

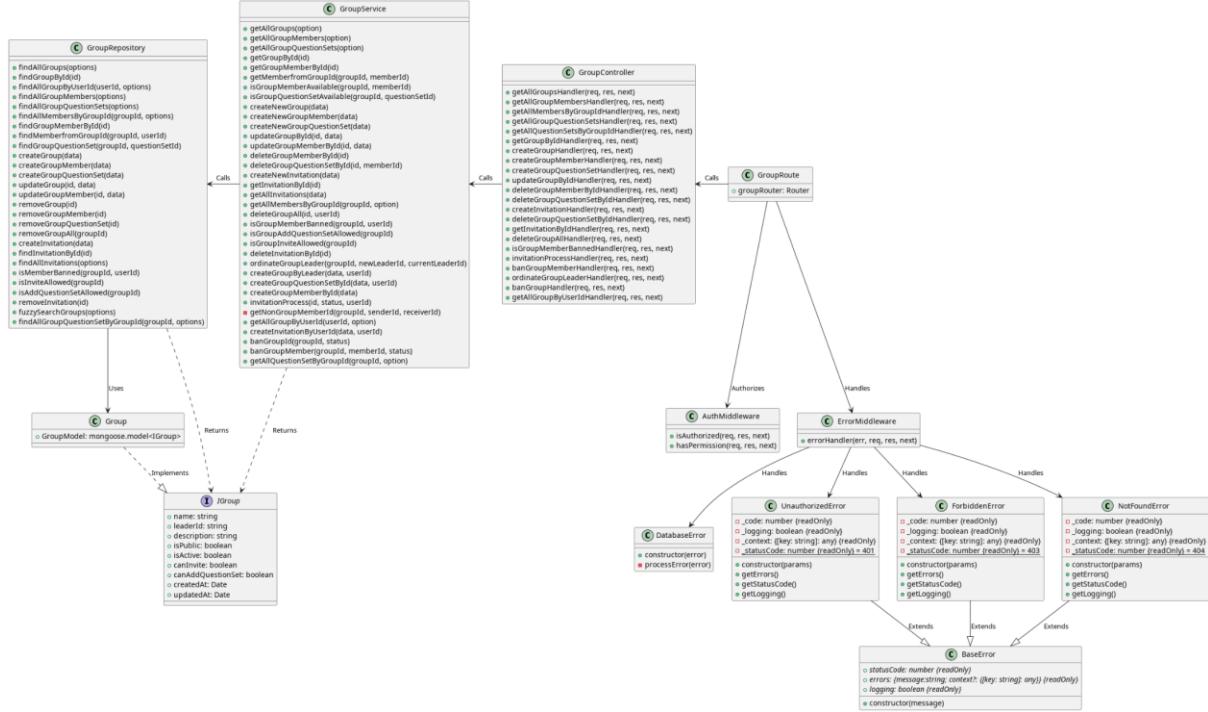


Figure 3.4.9.1.1 Block group¹⁰³

103 block groups.png

3.4.9.1.2 Unblock group

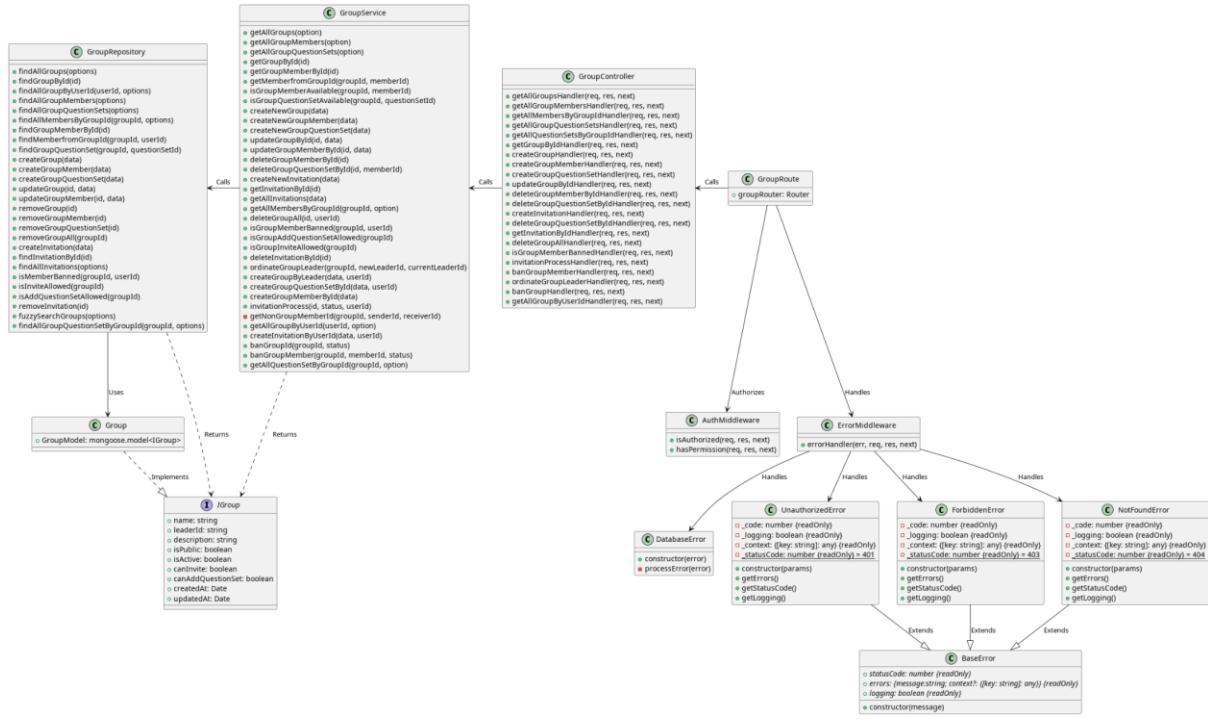


Figure 3.4.9.1.2 Unblock group¹⁰⁴

3.4.9.2 Sequence diagram

3.4.9.2.1 Block group

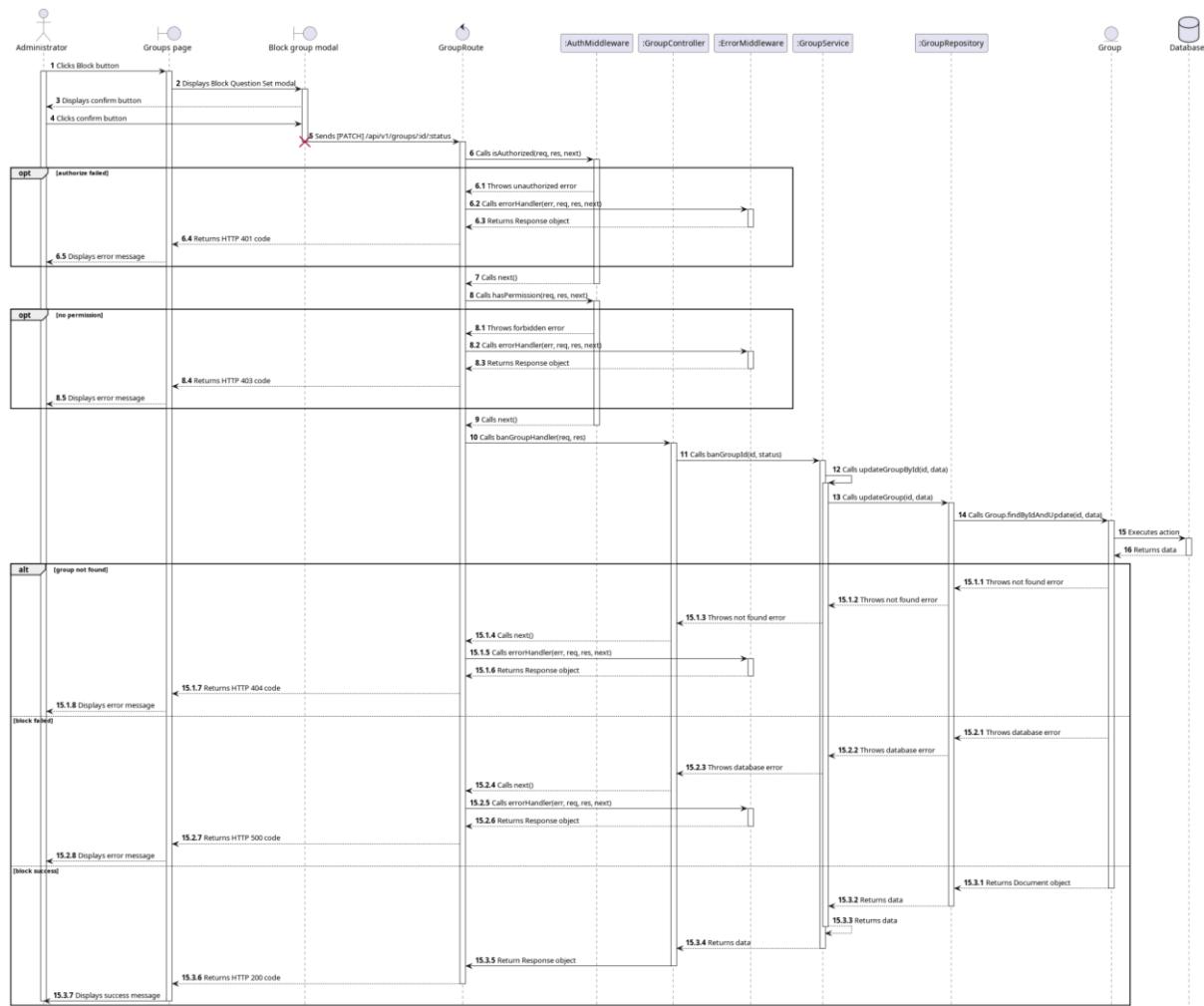


Figure 3.4.9.2.1 Block group¹⁰⁵

105 [block group.png](#)

3.4.9.2.2 Unblock group

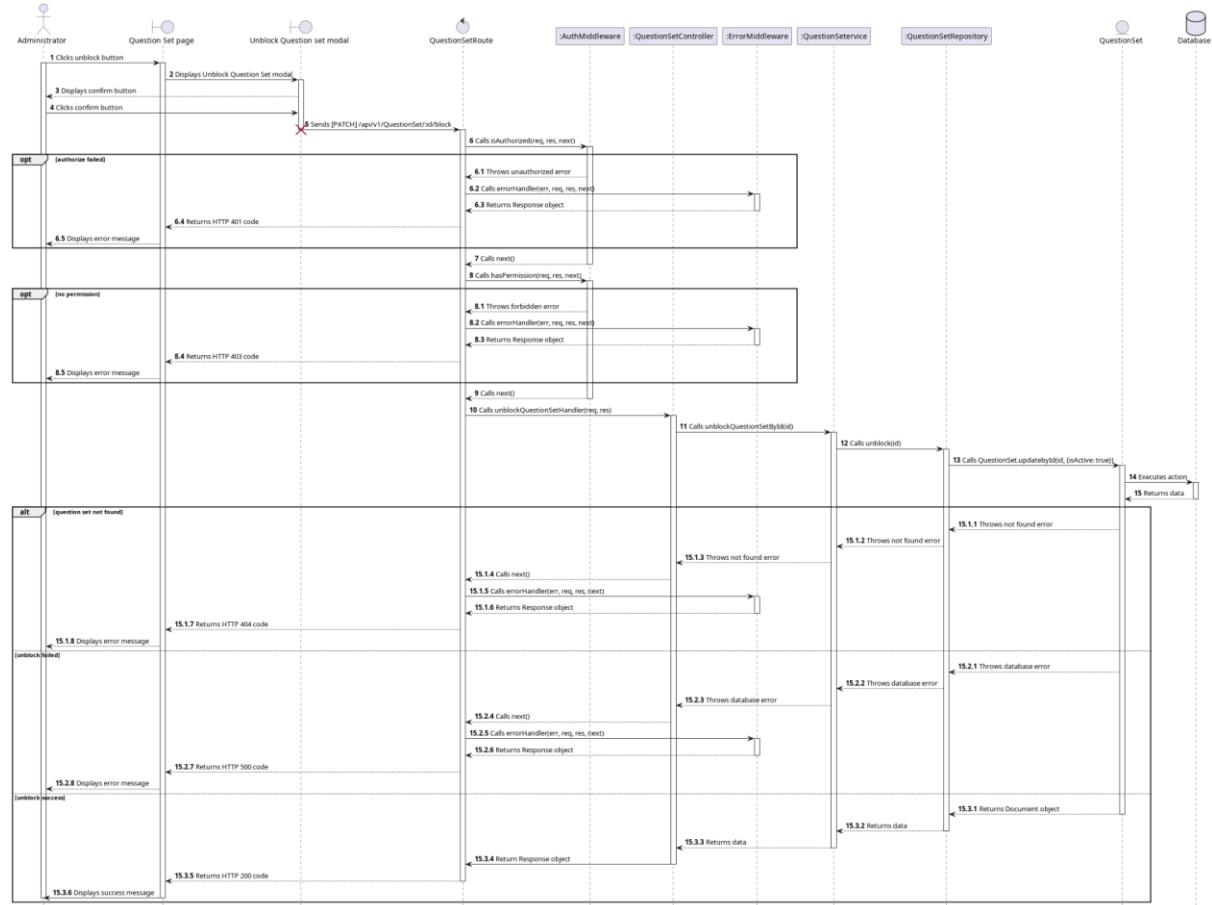


Figure 3.4.9.2.2 Unblock group¹⁰⁶

106 [unblock group.png](#)

3.5 Goal Management

3.5.1 View goals

3.5.1.1 Class diagram

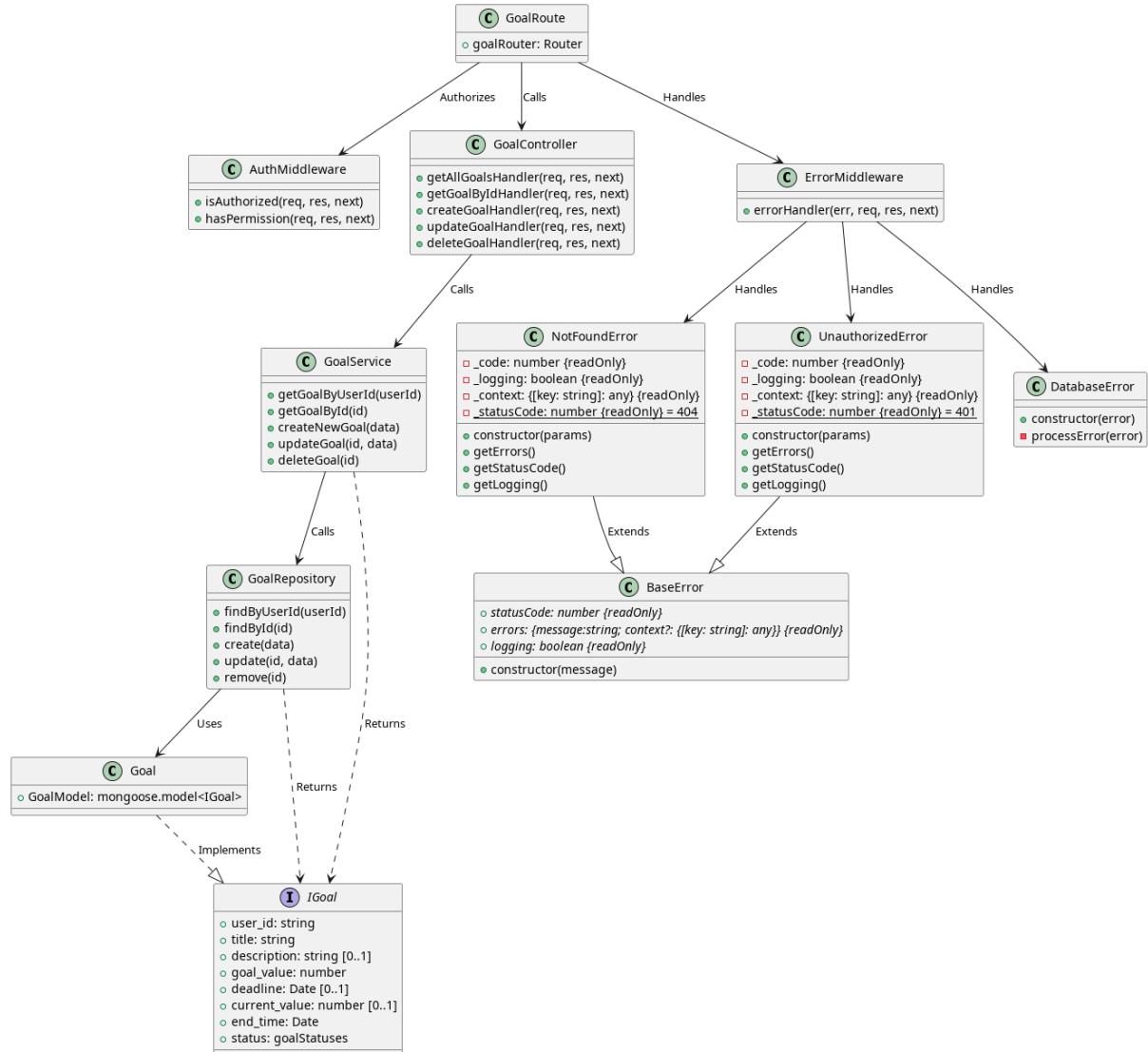


Figure 3.5.1.1: View goals¹⁰⁷

¹⁰⁷ [view goals.png](#)

3.5.1.2 Sequence diagram

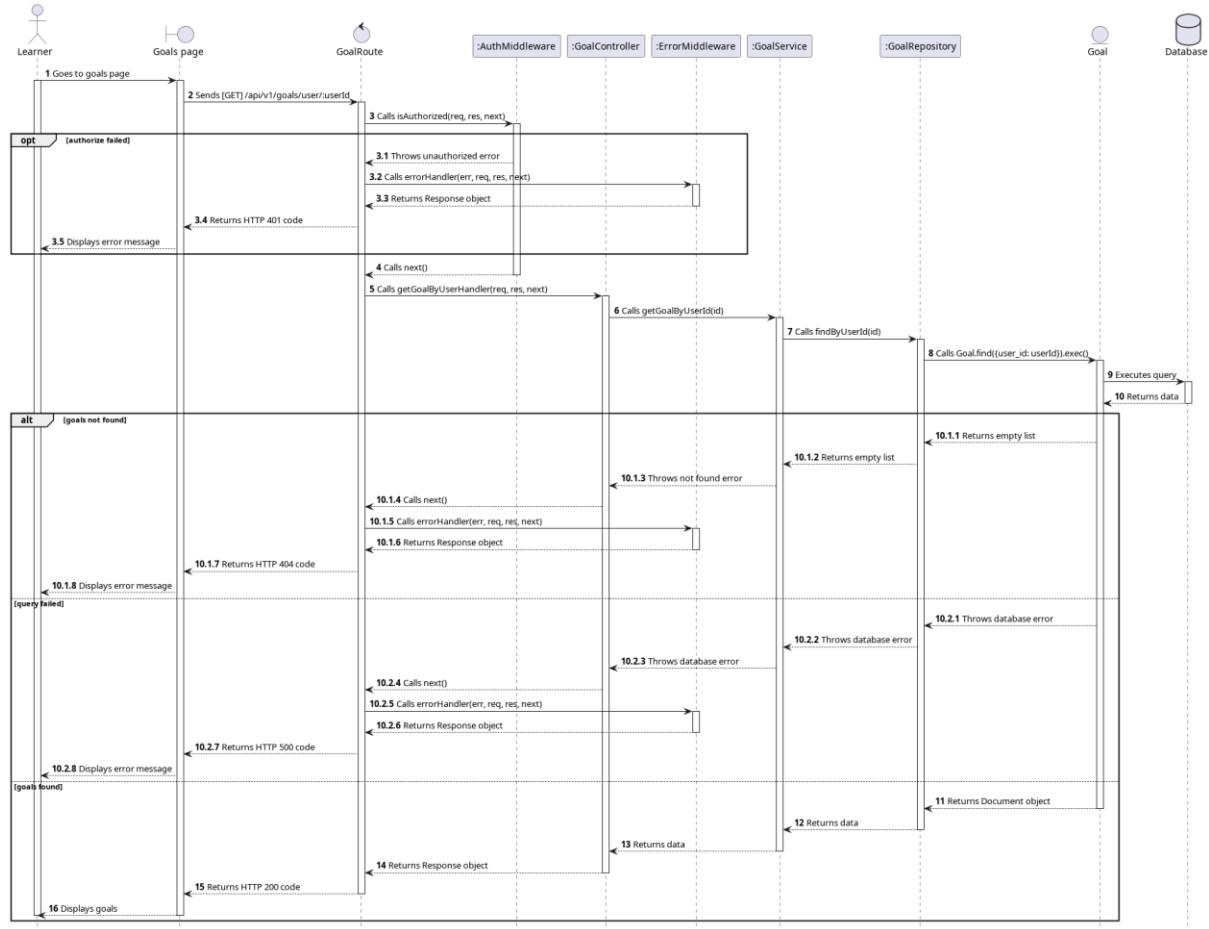


Figure 3.5.1.2: View goals¹⁰⁸

108 [view goals.png](#)

3.5.2 View goal detail

3.5.2.1 Class diagram

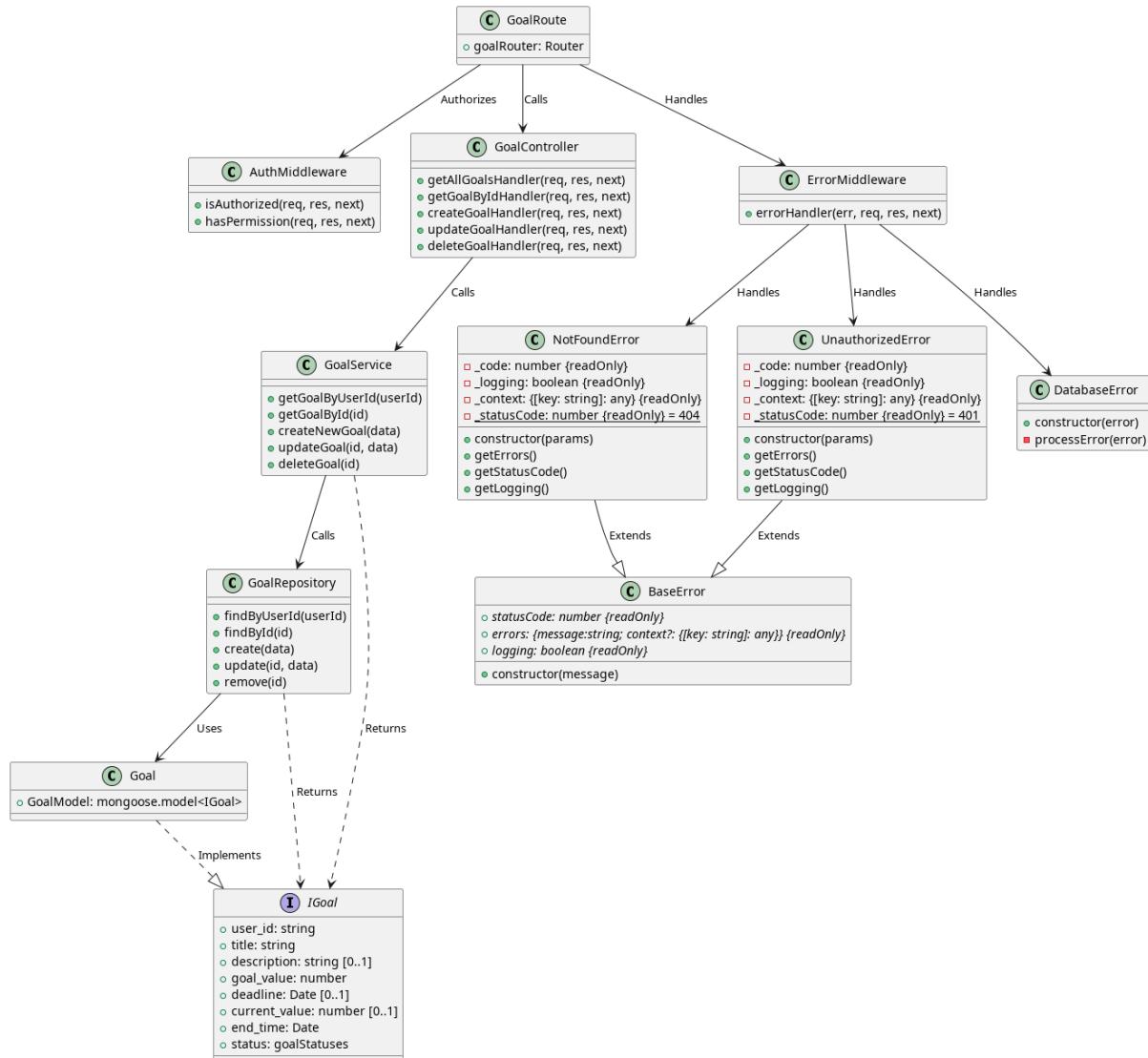


Figure 3.5.2.1: View goal detail¹⁰⁹

109 [view goals.png](#)

3.5.2.2 Sequence diagram

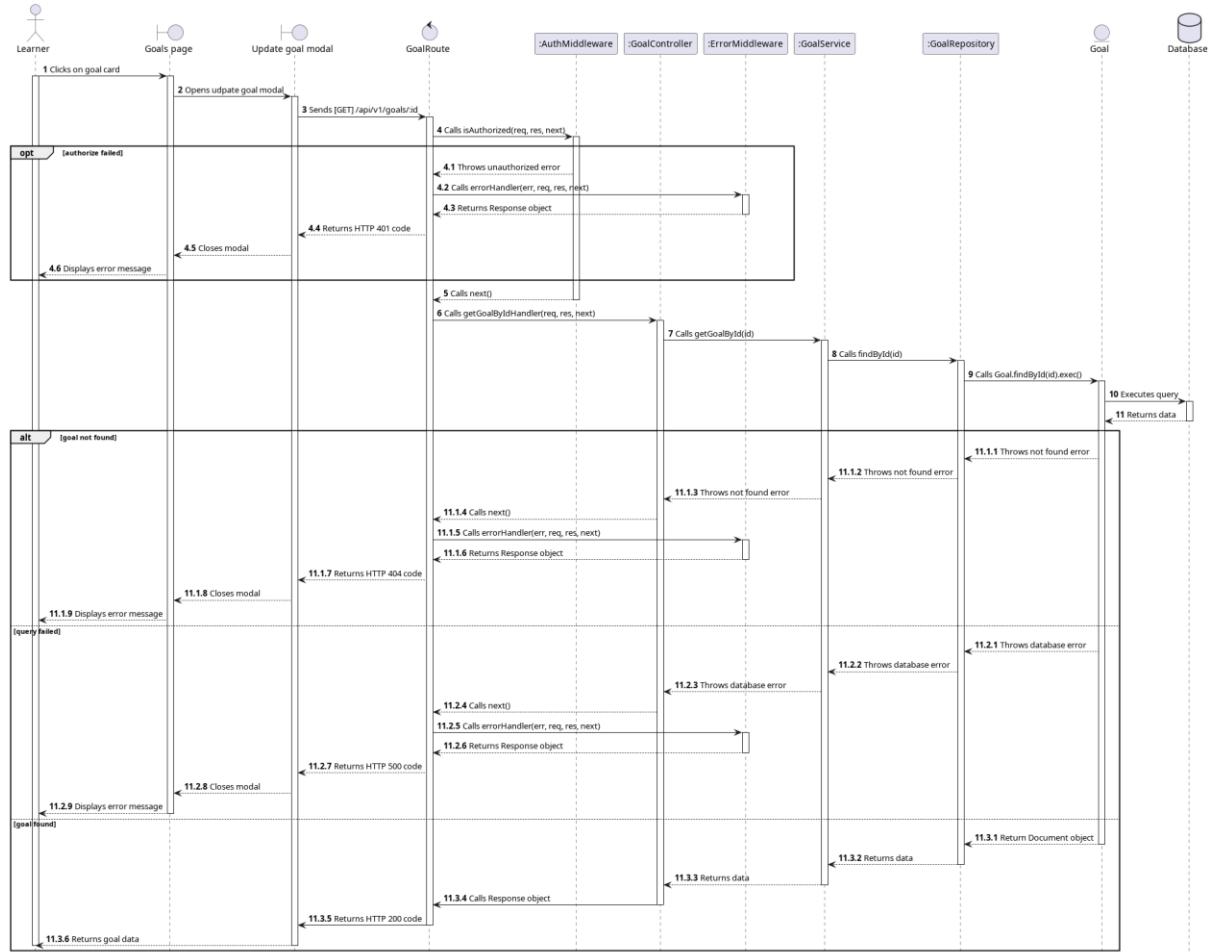


Figure 3.5.2.2: View goal detail¹¹⁰

¹¹⁰ [view goal detail.png](#)

3.5.3 Create goals

3.5.3.1 Class diagram

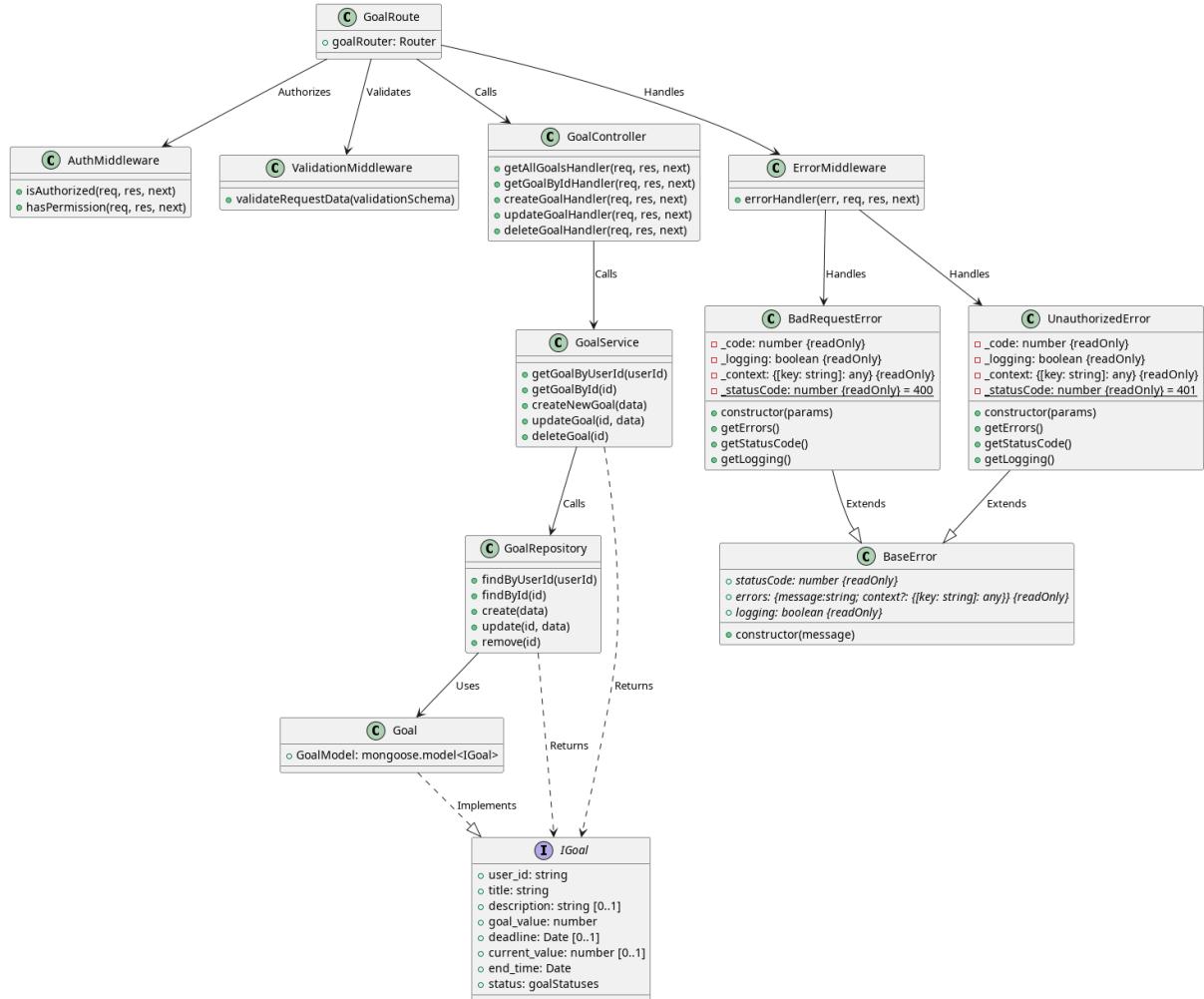


Figure 3.5.3.1: Create goals¹¹¹

¹¹¹ [view goals.png](#)

3.5.3.2 Sequence diagram

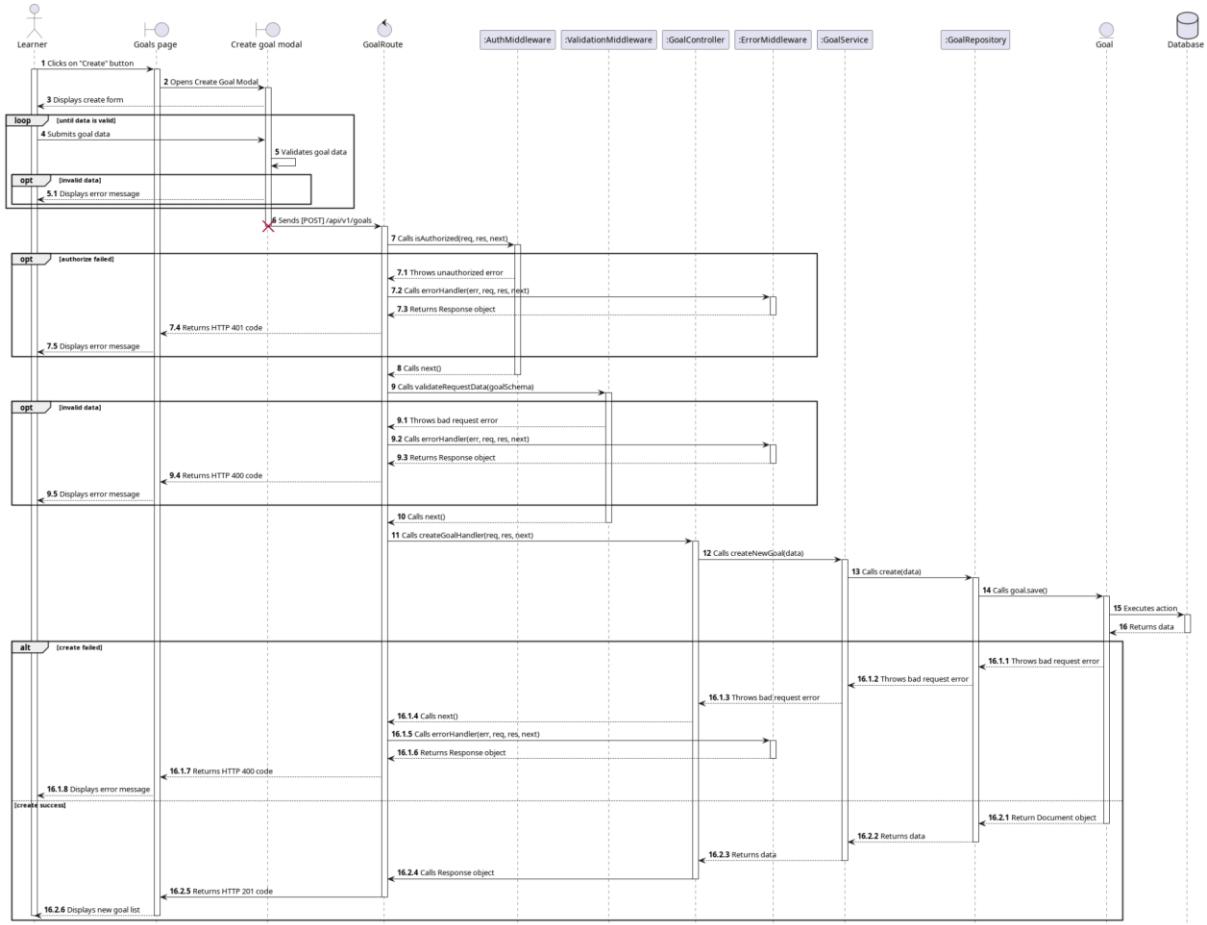


Figure 3.5.3.2: Create goals¹¹²

112 [create goal.png](#)

3.5.4 Update goal statuses

3.5.4.1 Class diagram

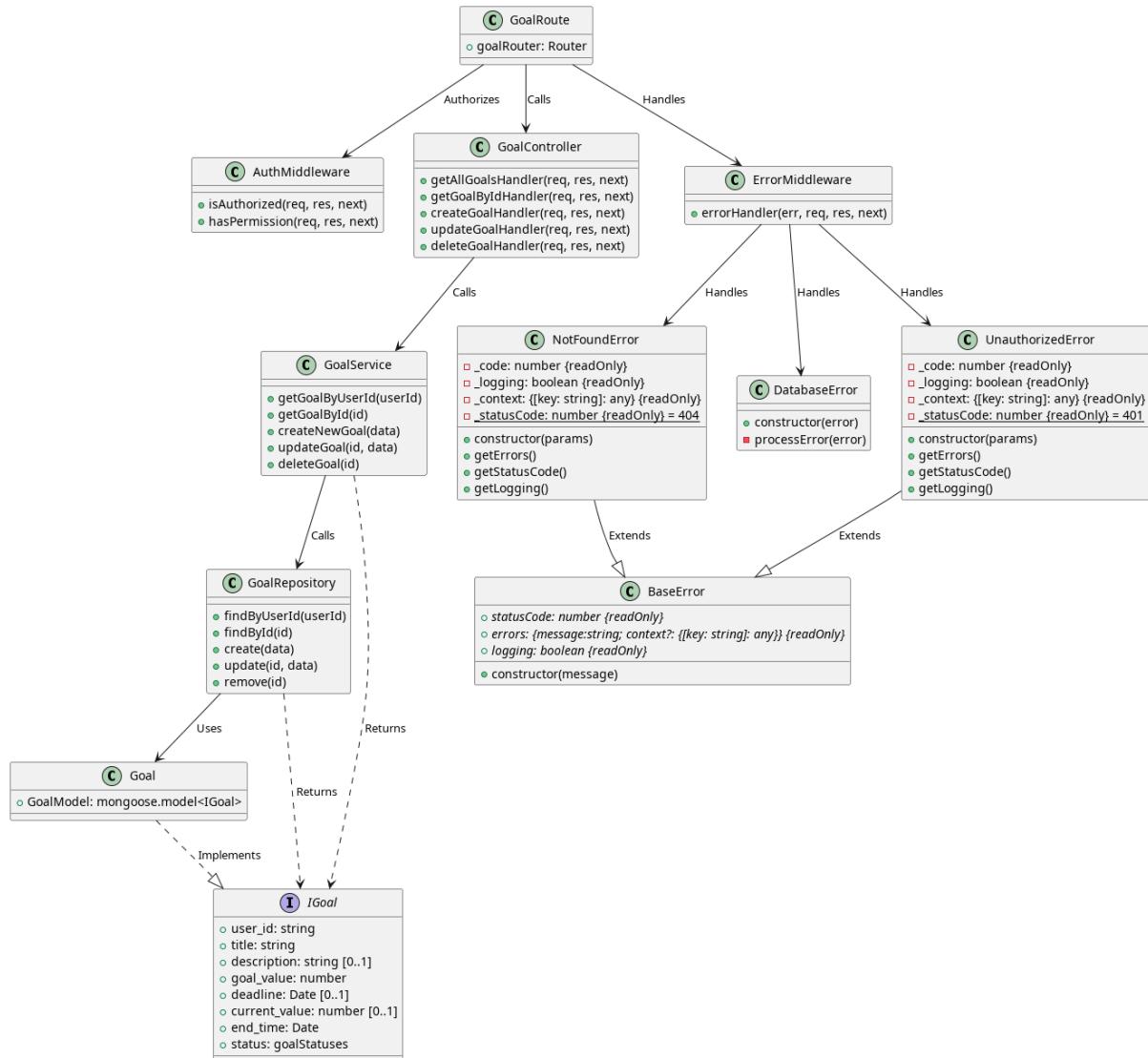


Figure 3.5.4.1: Update goal statuses¹¹³

¹¹³ [view goals.png](#)

3.5.4.2 Sequence diagram

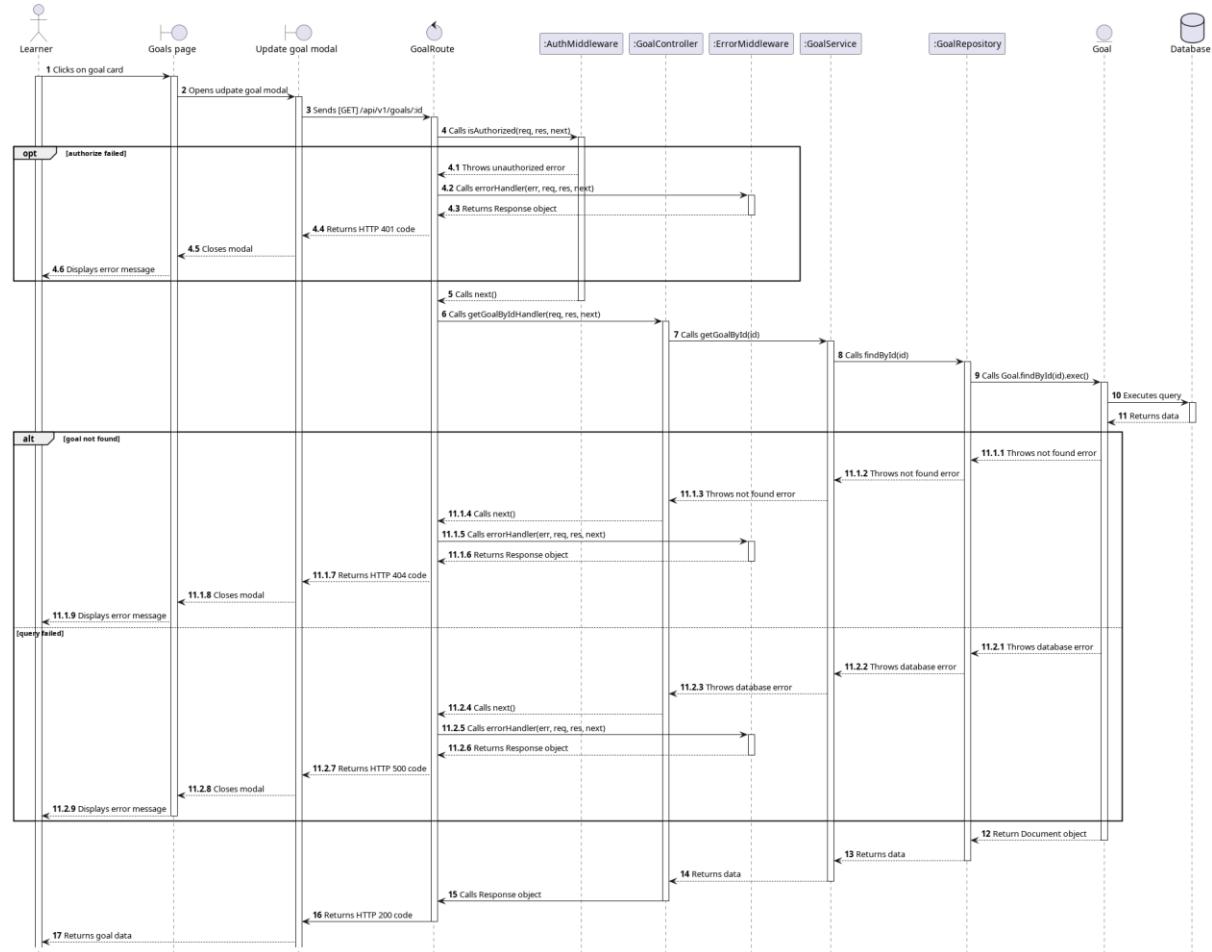


Figure 3.5.4.2.a: View goal detail¹¹⁴

¹¹⁴ [view goal detail.png](#)

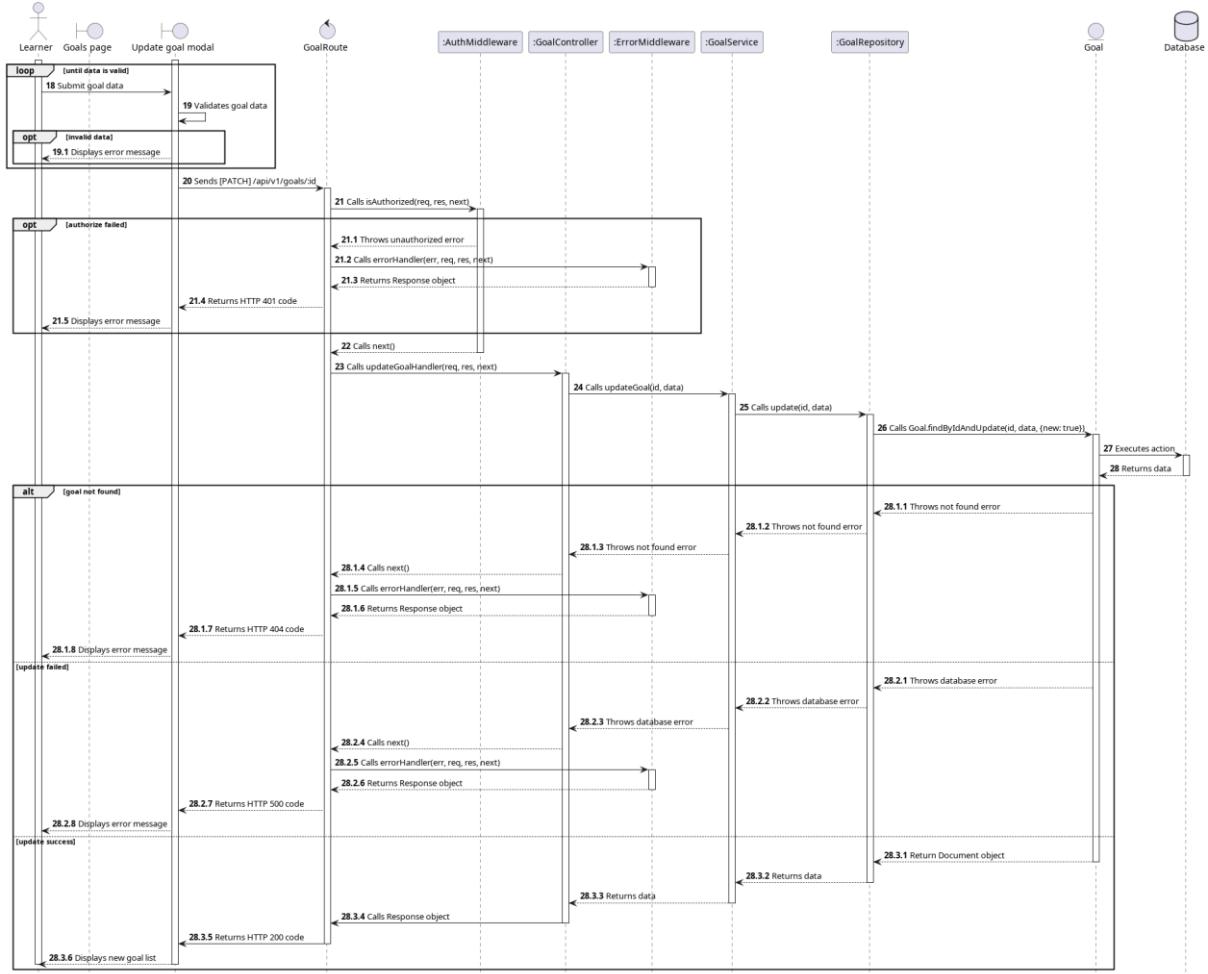


Figure 3.5.4.2.b: Update goal statuses¹¹⁵

115 [update goal.png](#)

3.5.5 Delete goals

3.5.5.1 Class diagram

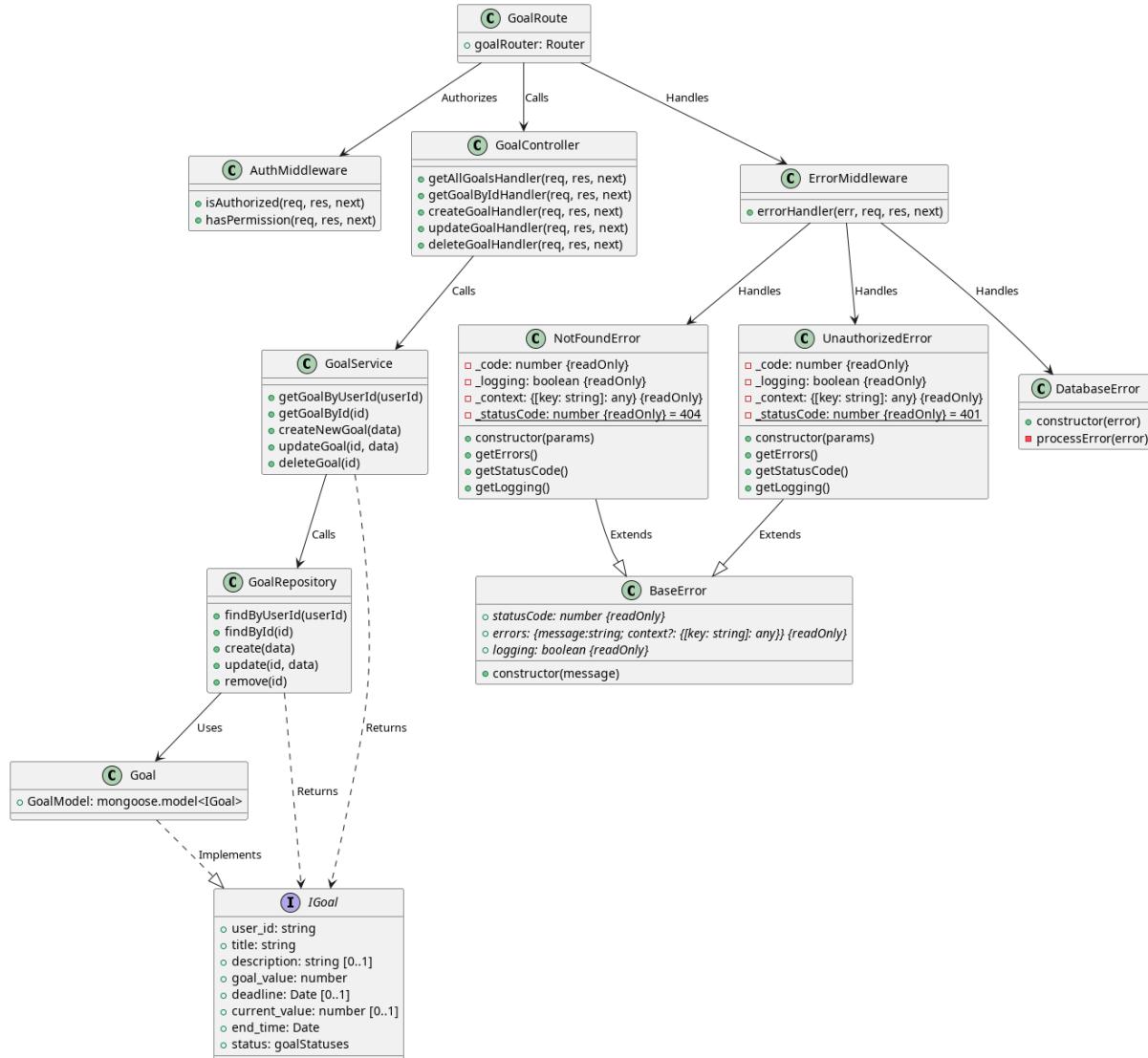


Figure 3.5.5.1: Delete goals¹¹⁶

¹¹⁶ [delete goal.png](#)

3.5.5.2 Sequence diagram

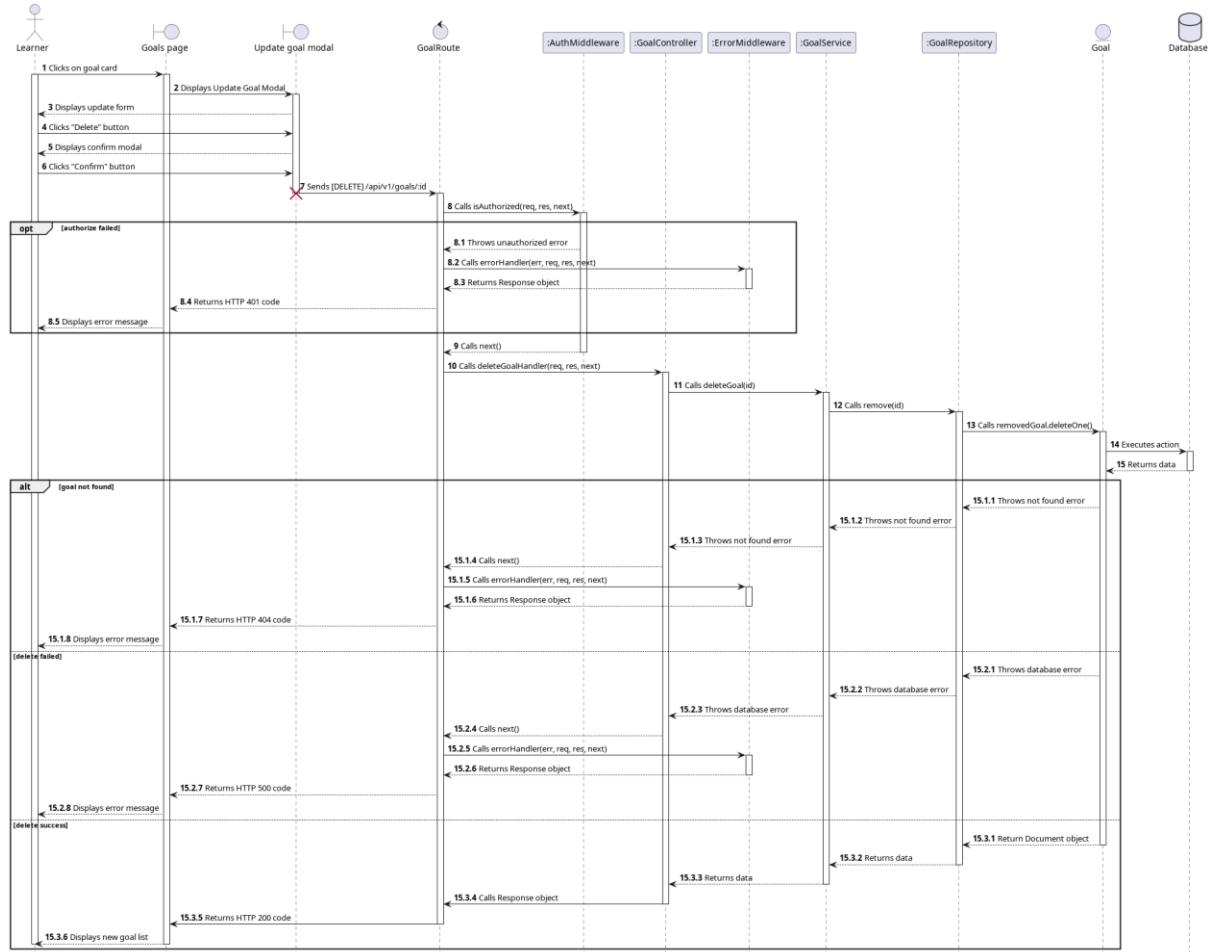


Figure 3.5.5.2: Delete goals¹¹⁷

¹¹⁷ [delete goal.png](#)

3.6 Settings Management

3.6.1 View settings

3.6.1.1 Class diagram

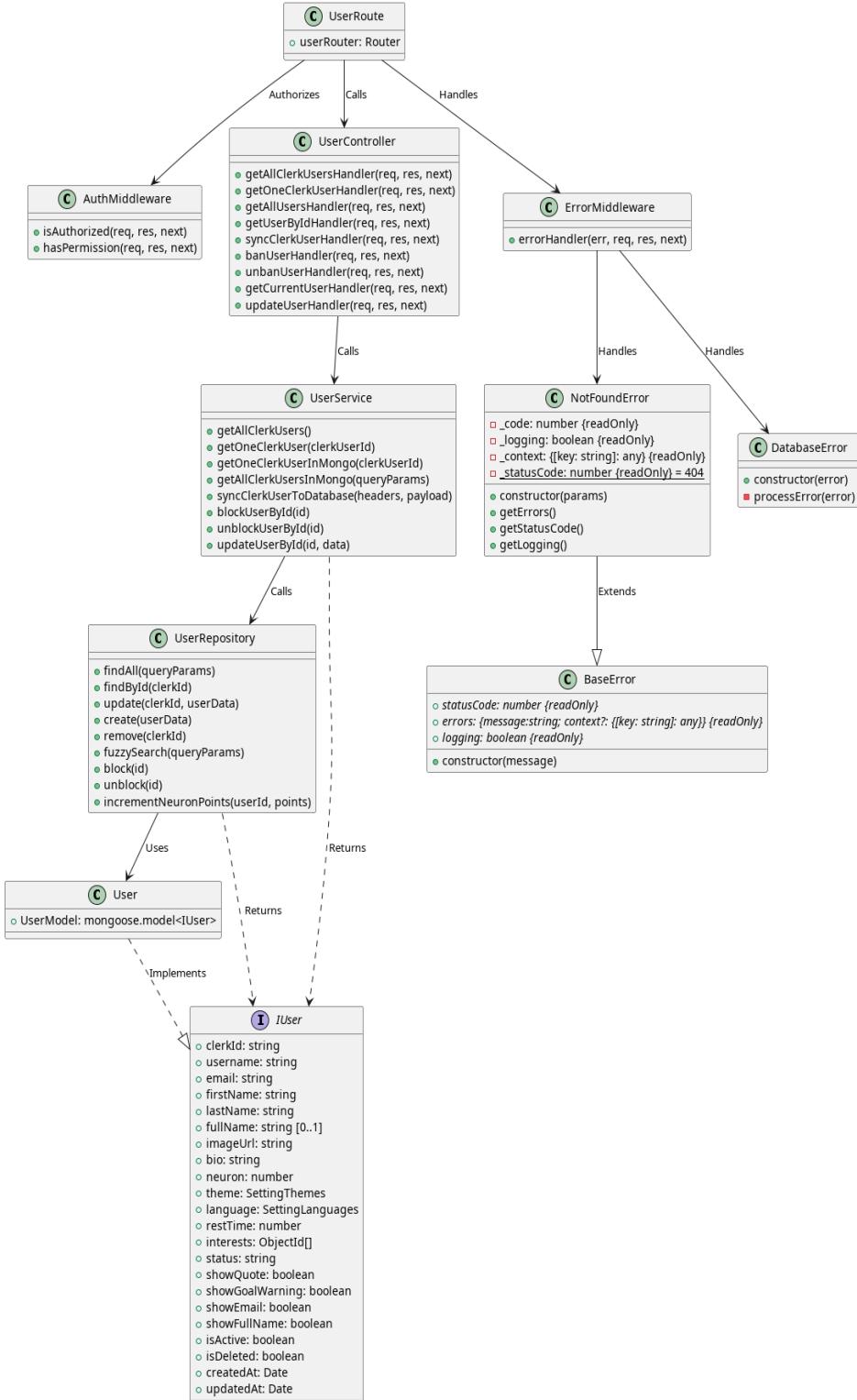


Figure 3.6.1.1: View settings¹¹⁸

118 [view settings.png](#)

3.6.1.2 Sequence diagram

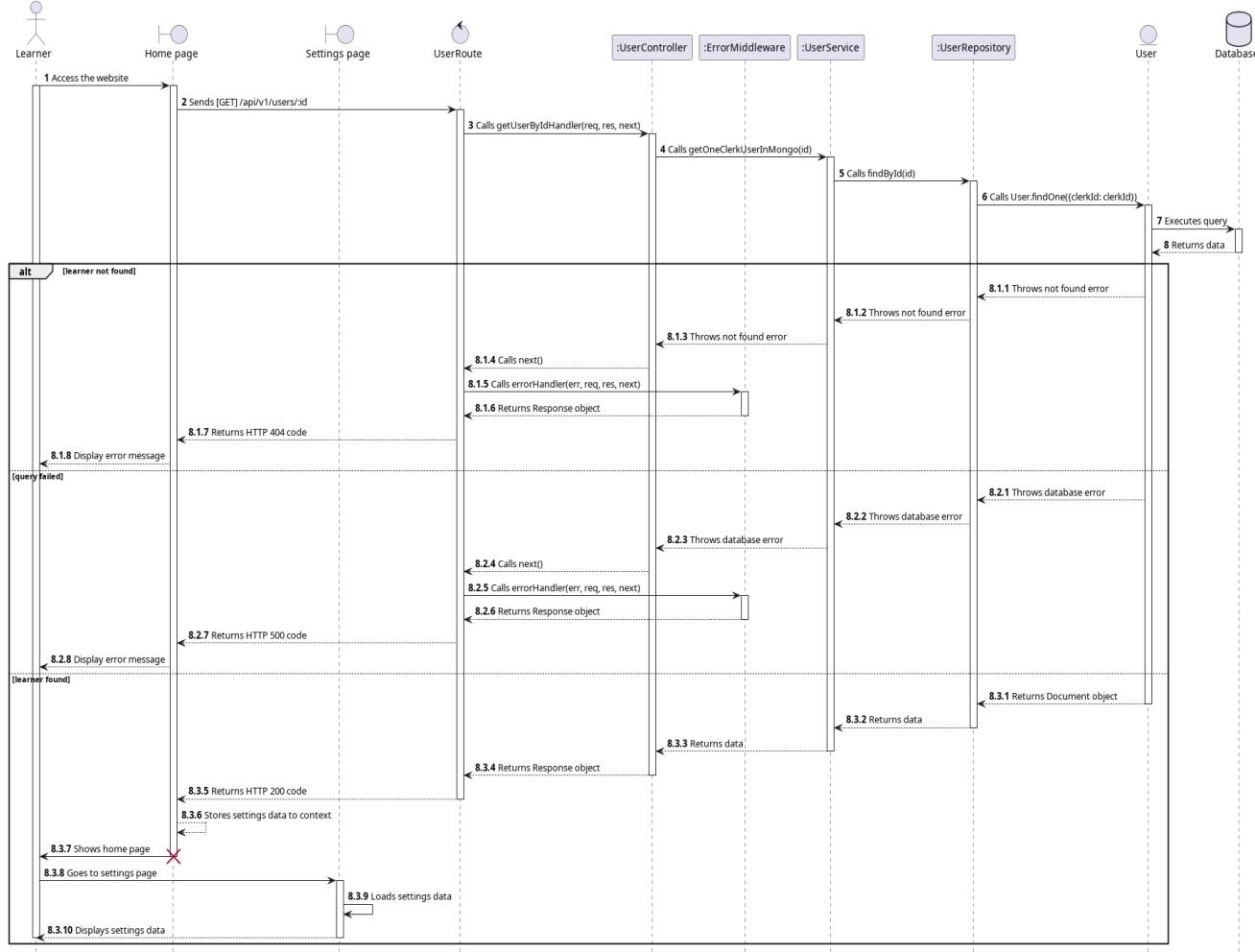


Figure 3.6.1.2: View settings¹¹⁹

119 [view settings.png](#)

3.6.2 Update system appearance

3.6.2.1 Class diagram

3.6.2.1.1 Change theme

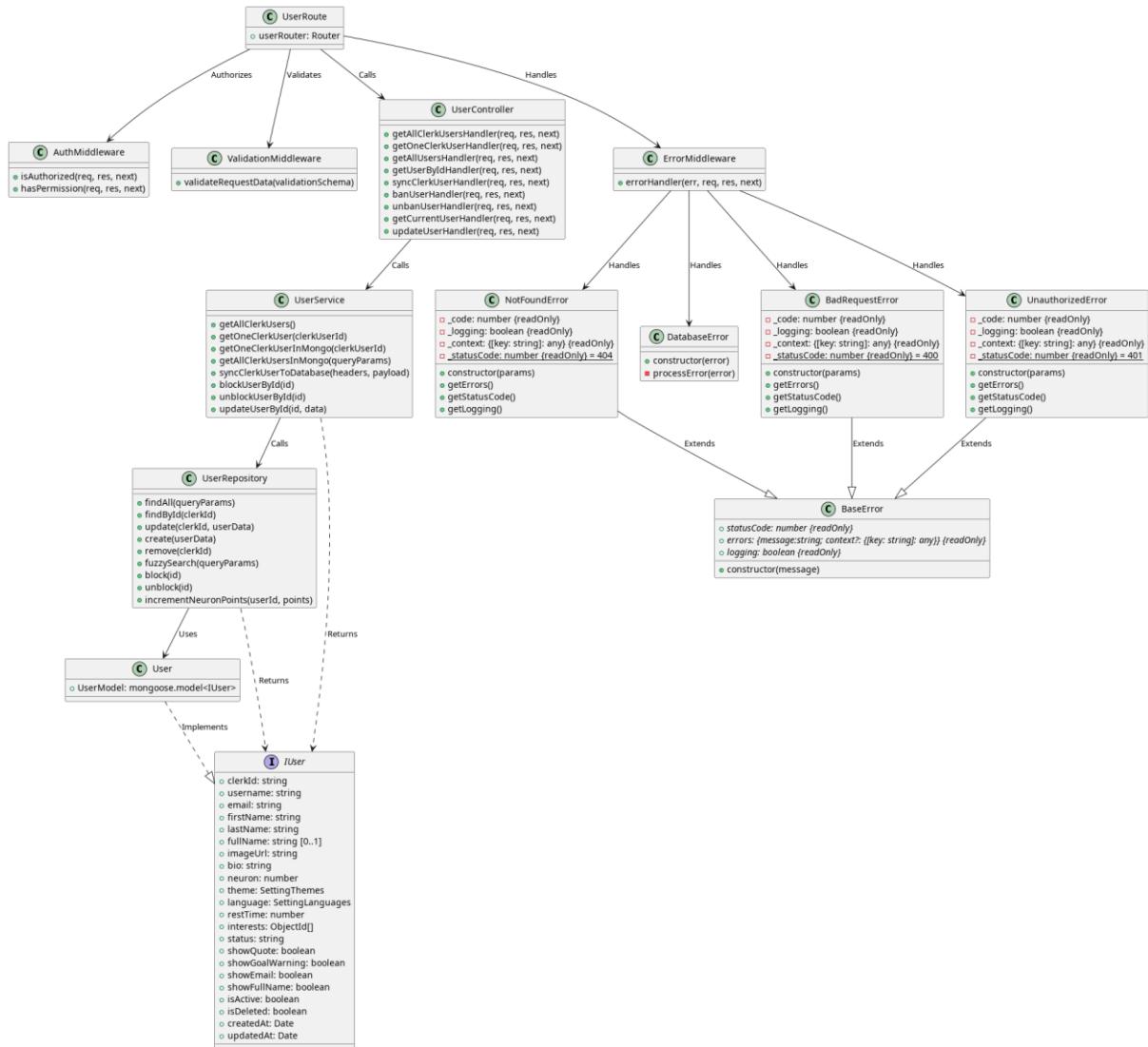


Figure 3.6.2.1.1: Change theme¹²⁰

¹²⁰ change themes.png

3.6.2.1.2 Change language

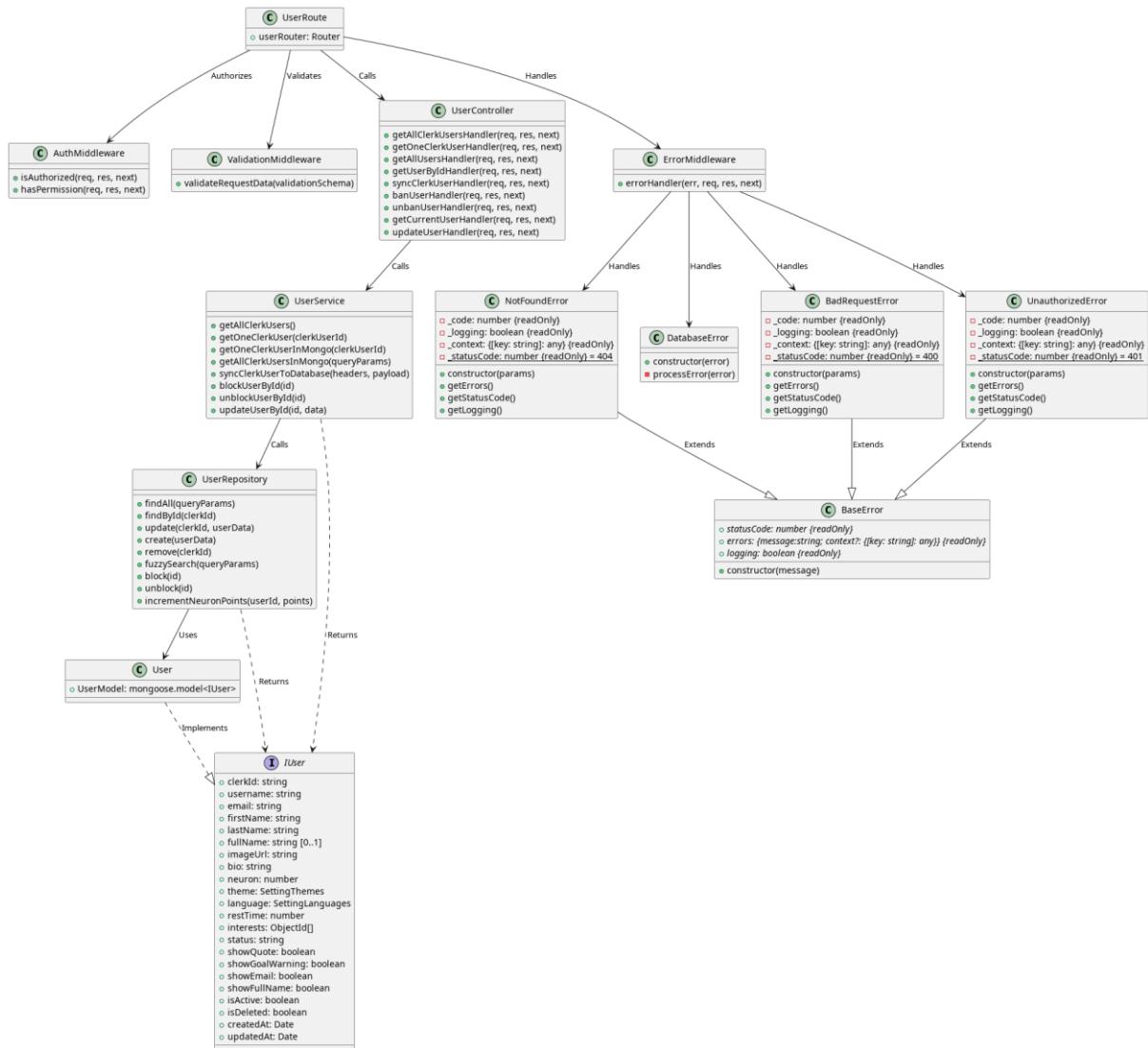


Figure 3.6.2.1.1: Change language¹²¹

¹²¹ [change languages.png](#)

3.6.2.1.3 Toggle component

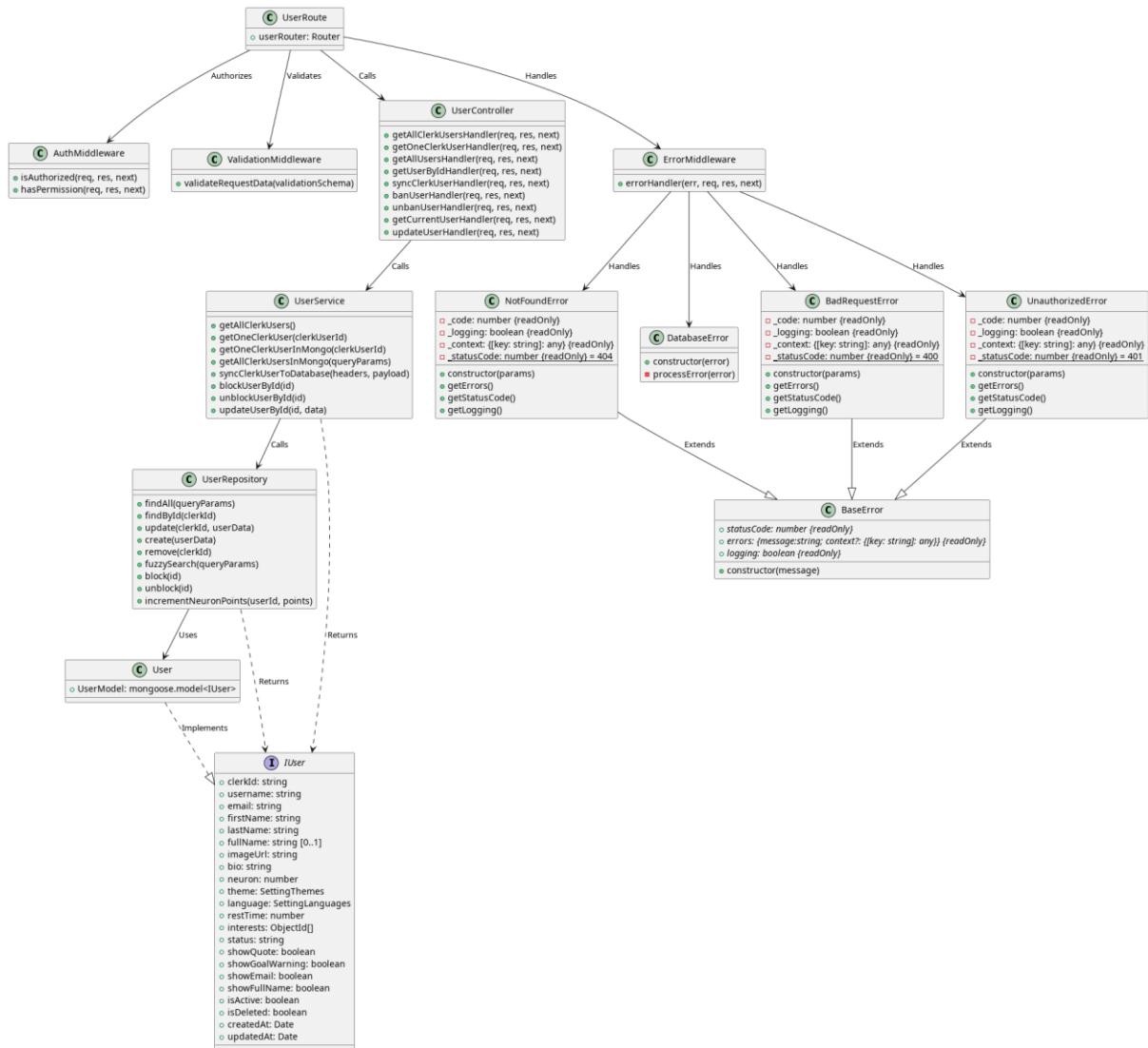


Figure 3.6.2.1.3: Toggle component¹²²

¹²² [toggle component.png](#)

3.6.2.2 Sequence diagram

3.6.2.2.1 Change theme

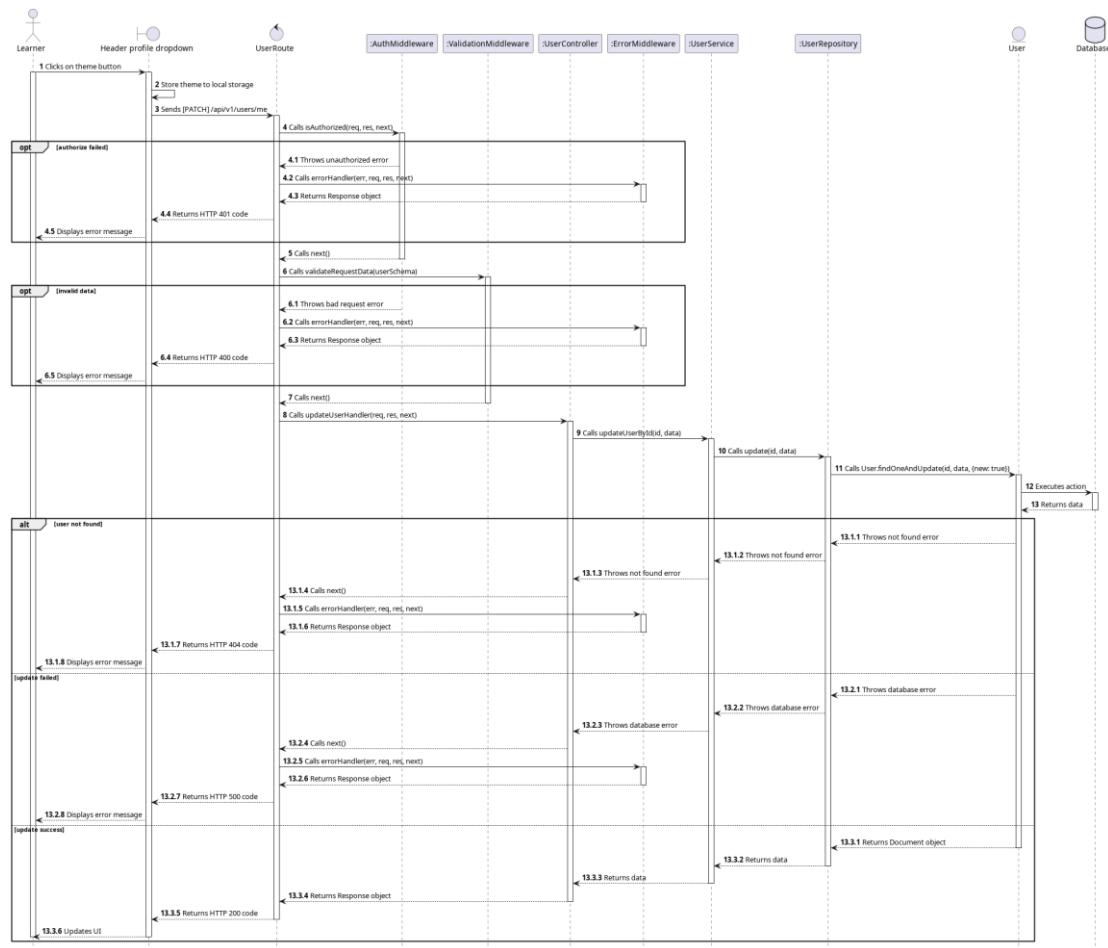


Figure 3.6.2.2.1.a: Learner change theme¹²³

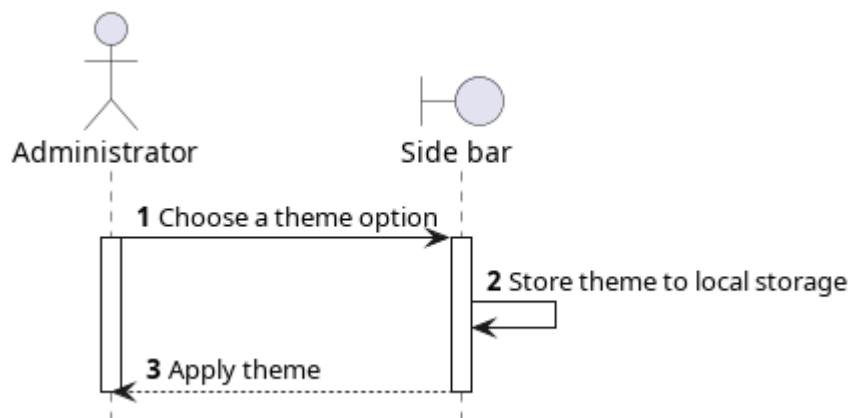


Figure 3.6.2.2.1.b: Administrator change theme

¹²³ [change theme.png](#)

3.6.2.2.2 Change language

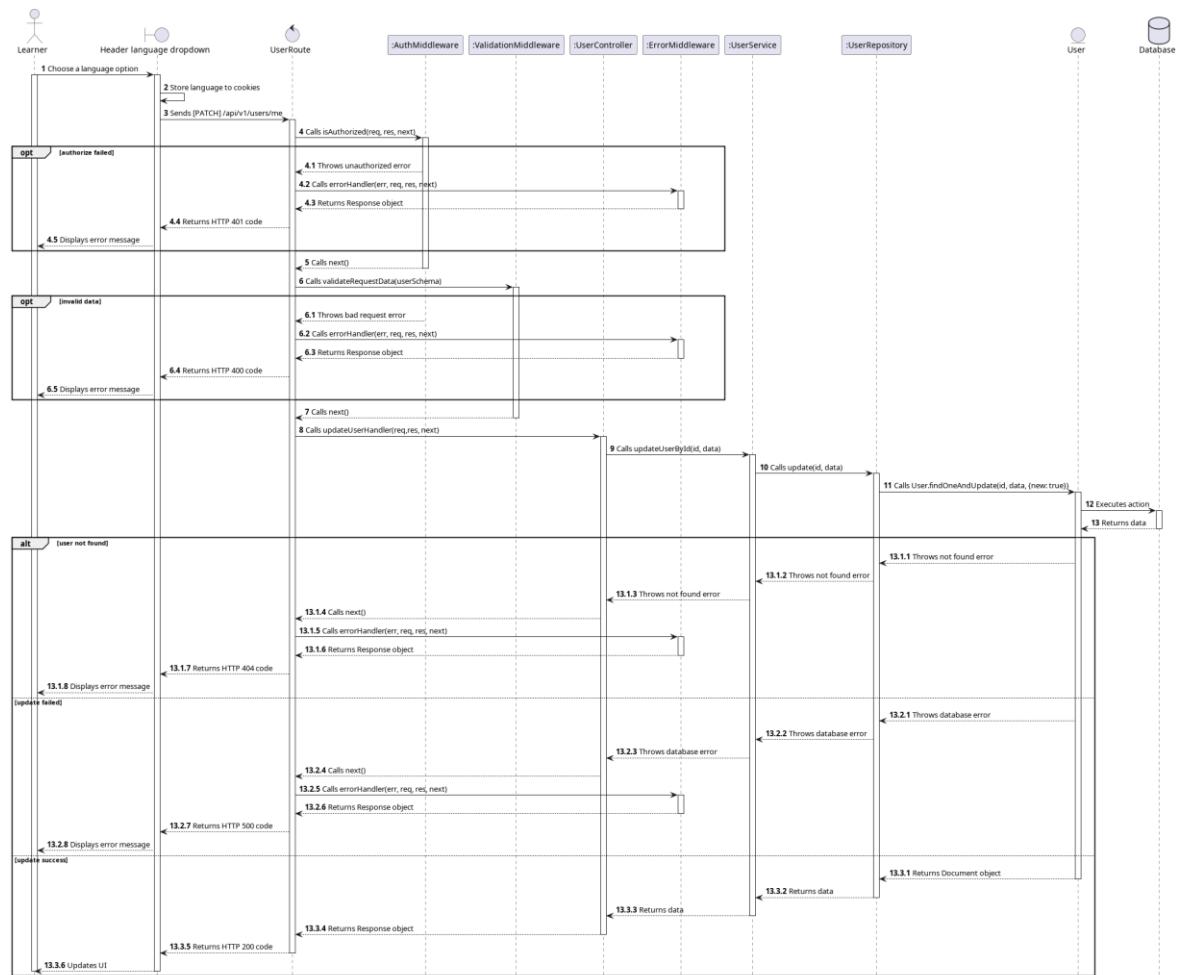


Figure 3.6.2.2.2.a: Learner change language¹²⁴

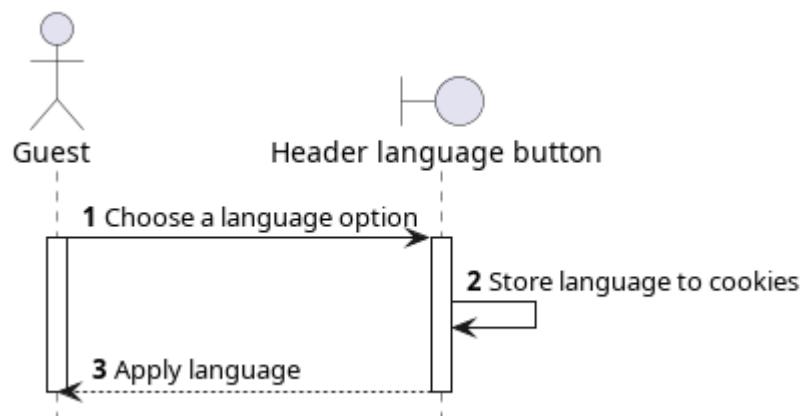


Figure 3.6.2.2.2.b: Guest change language

¹²⁴ [change language.png](#)

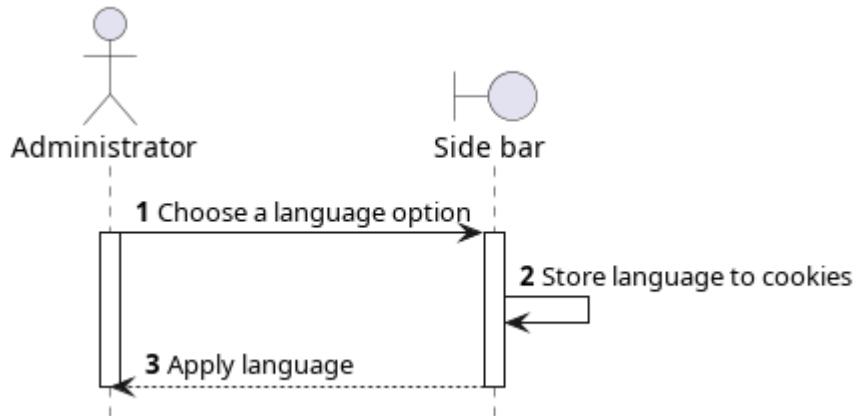


Figure 3.6.2.2.2.c: Administrator change language

3.6.2.2.3 Toggle component

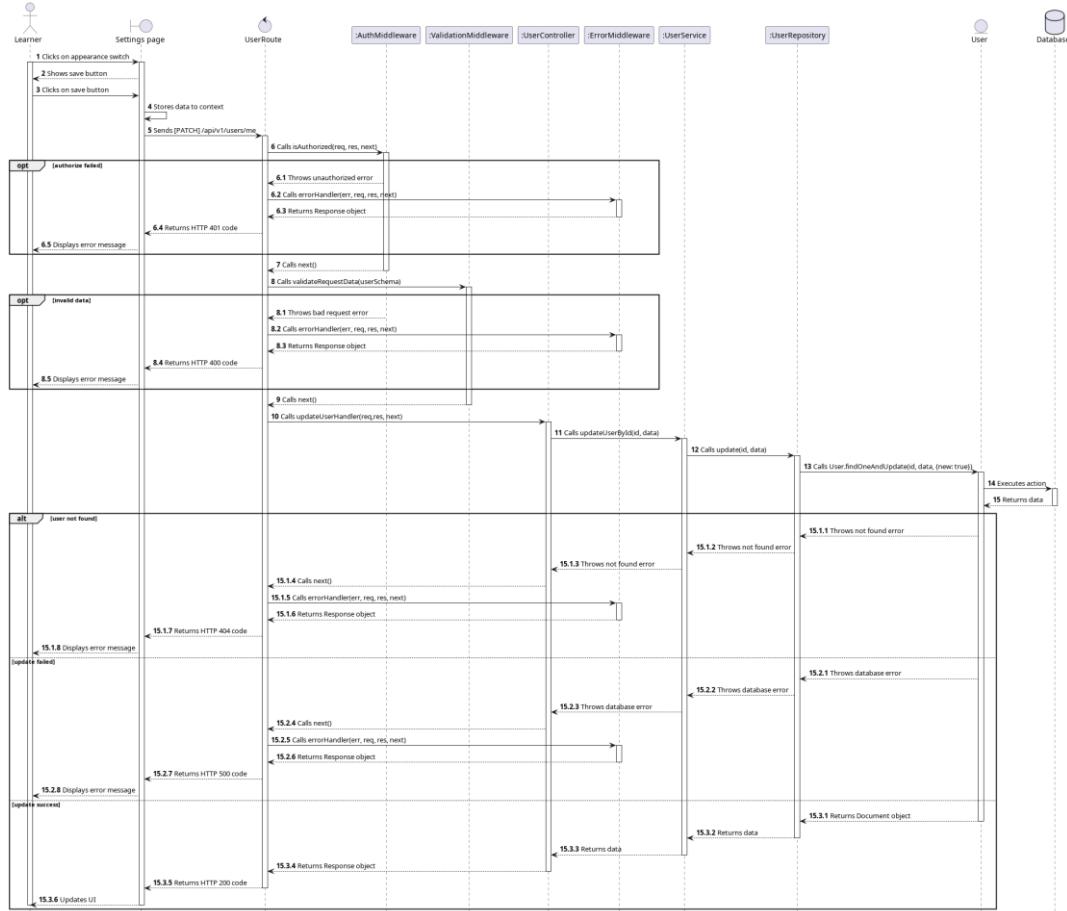


Figure 3.6.2.2.3: Toggle component¹²⁵

¹²⁵ [toggle component.png](#)

3.6.3 Set rest reminder

3.6.3.1 Class diagram

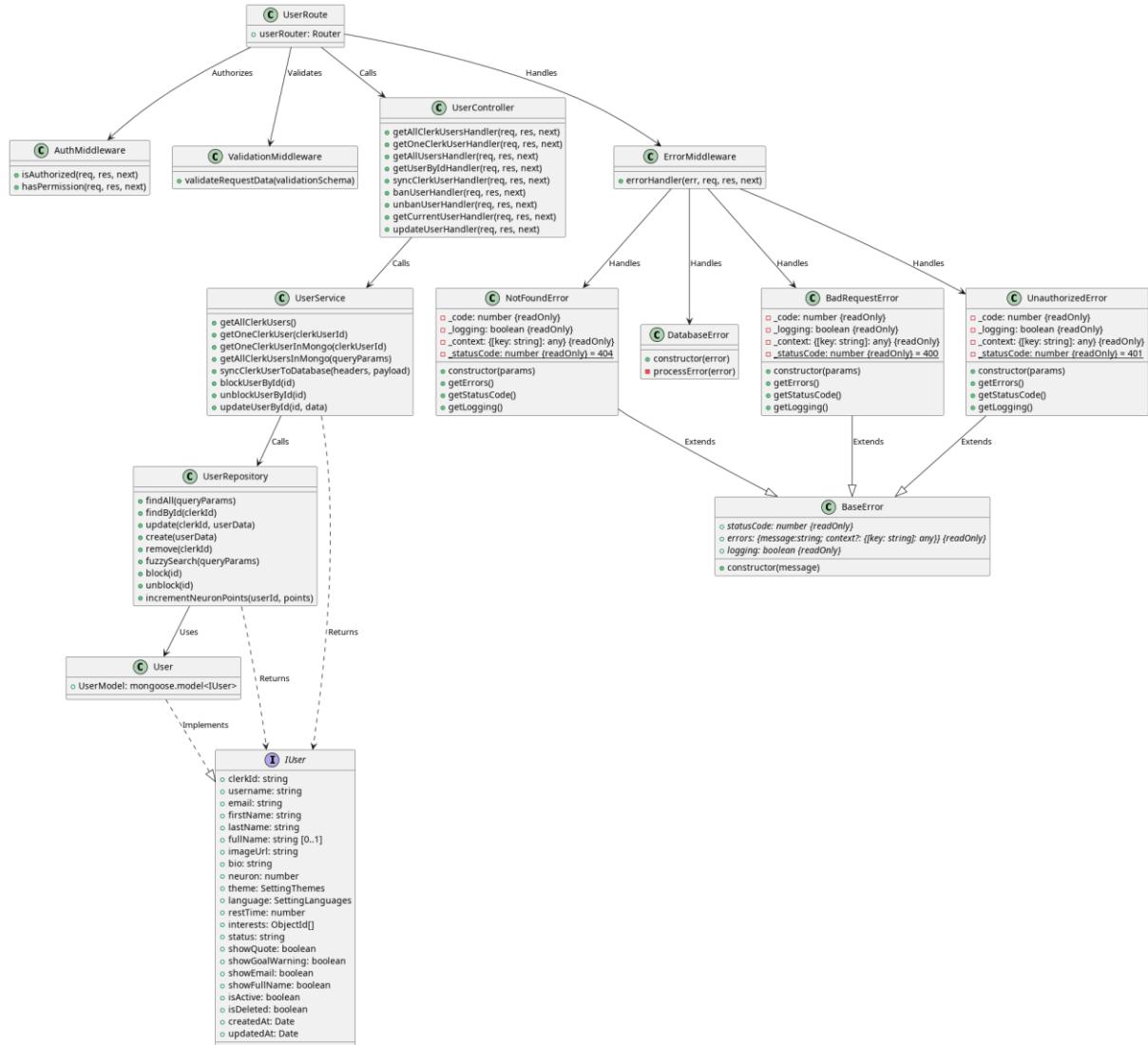


Figure 3.6.3.1: Set rest reminder¹²⁶

¹²⁶ [set rest reminder.png](#)

3.6.3.2 Sequence diagram

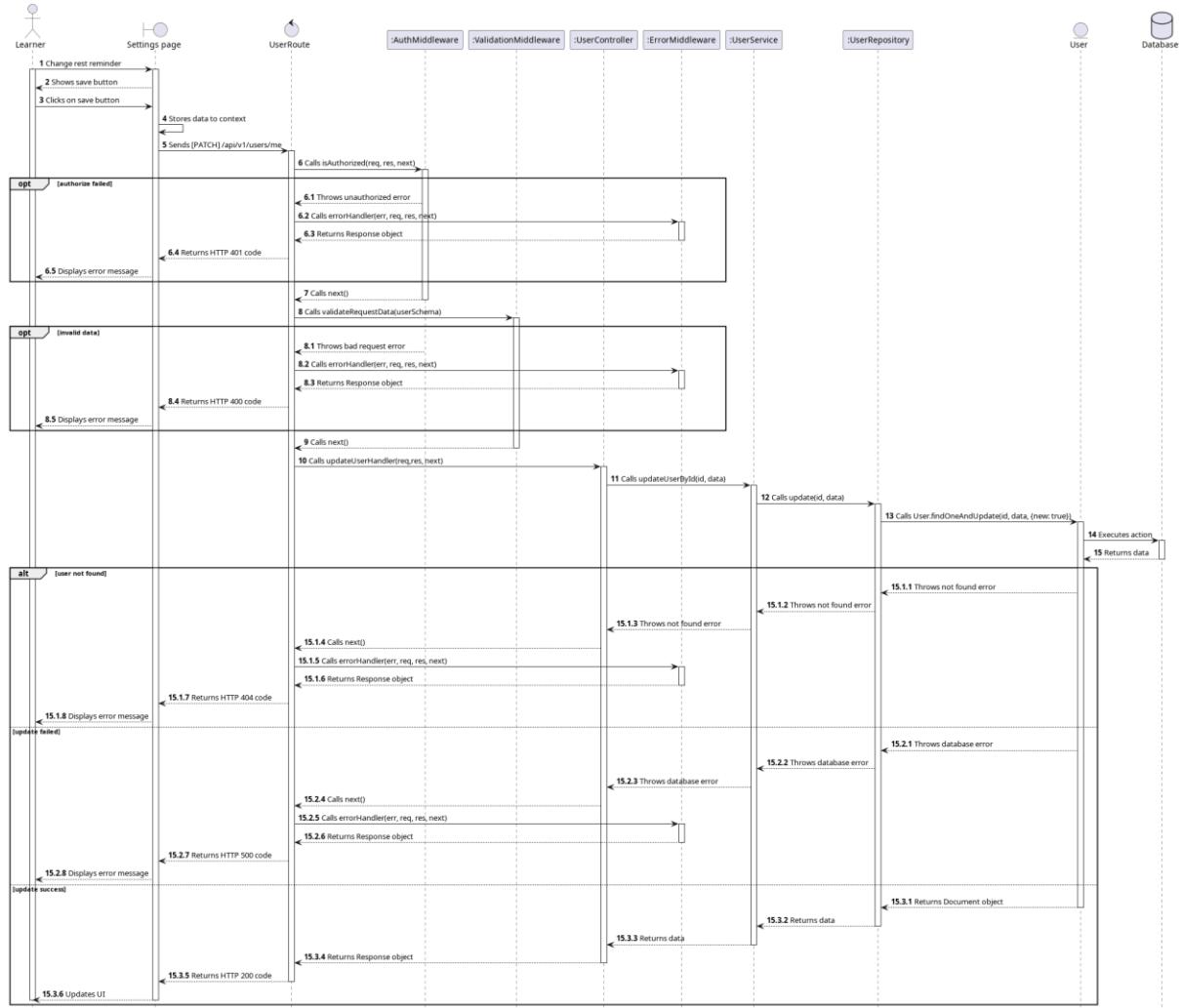


Figure 3.6.3.2: Set rest reminder¹²⁷

¹²⁷ [set rest reminder.png](#)

3.6.4 Reset setting

3.6.4.1 Class diagram

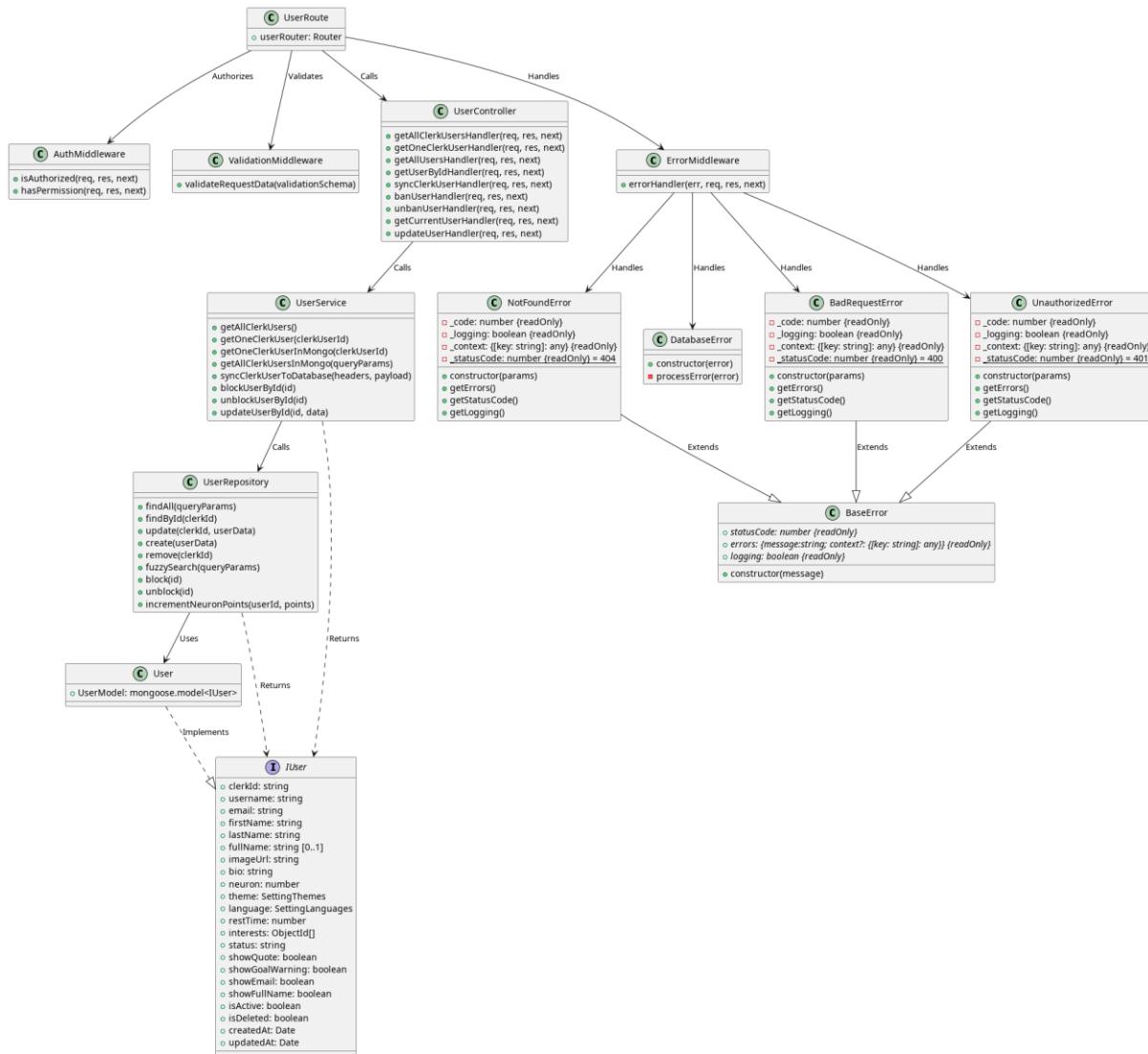


Figure 3.6.4.1: Reset setting¹²⁸

¹²⁸ [reset settings.png](#)

3.6.4.2 Sequence diagram

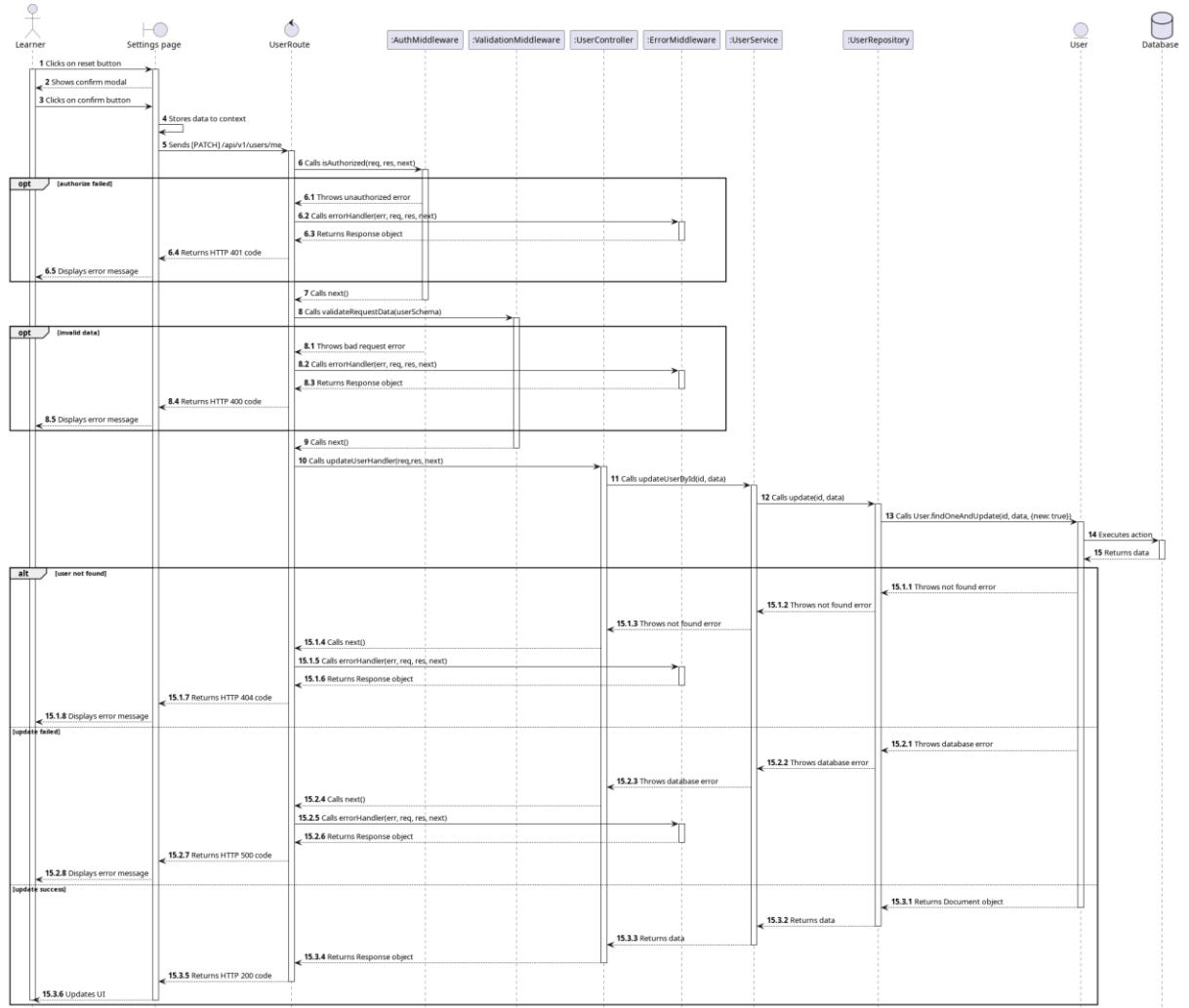


Figure 3.6.4.2: Reset setting¹²⁹

¹²⁹ reset settings.png

3.7 Learner Management

3.7.1 View learners

3.7.1.1 Class diagram

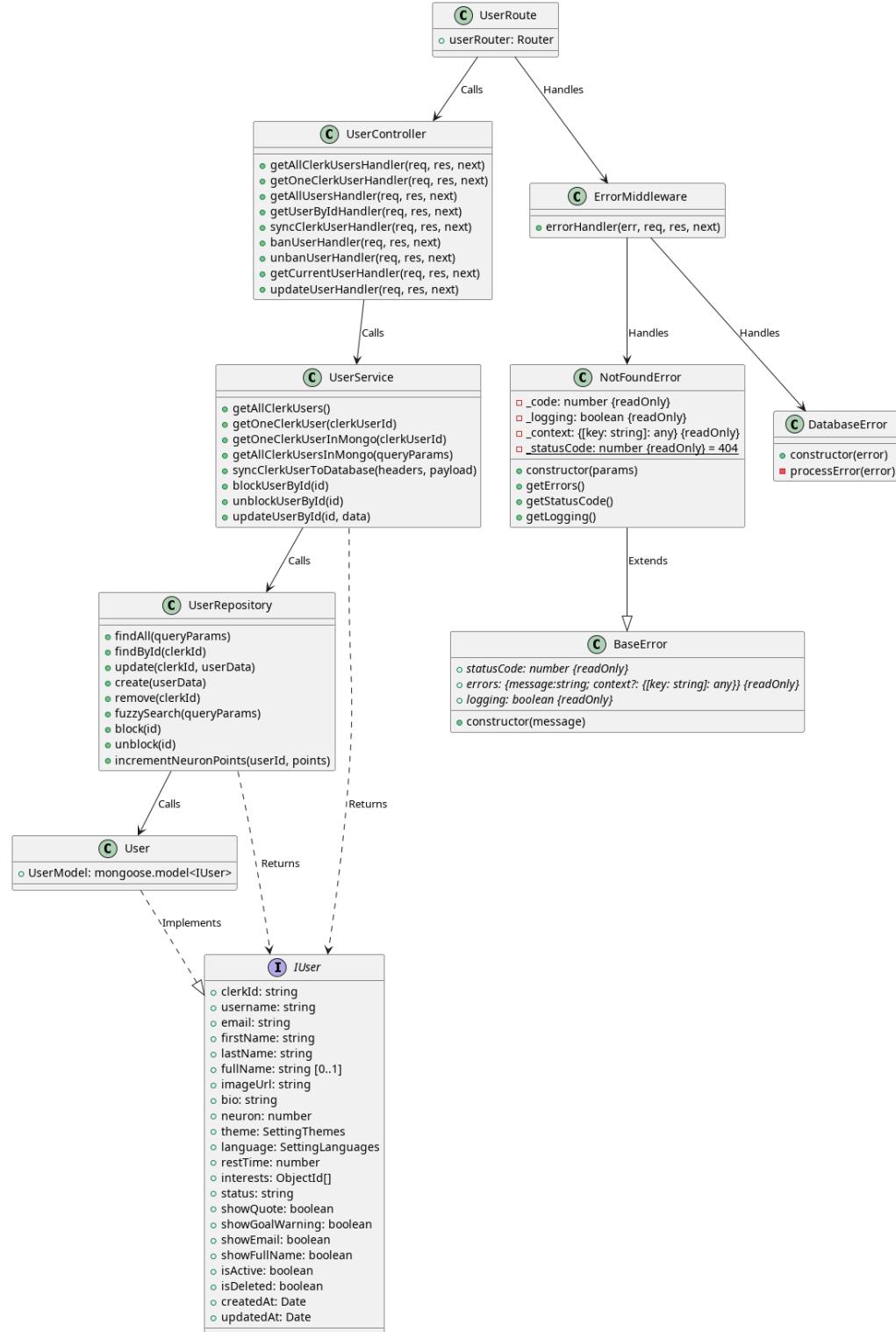


Figure 3.7.1.1: View learners¹³⁰

¹³⁰ [view learners.png](#)

3.7.1.2 Sequence diagram

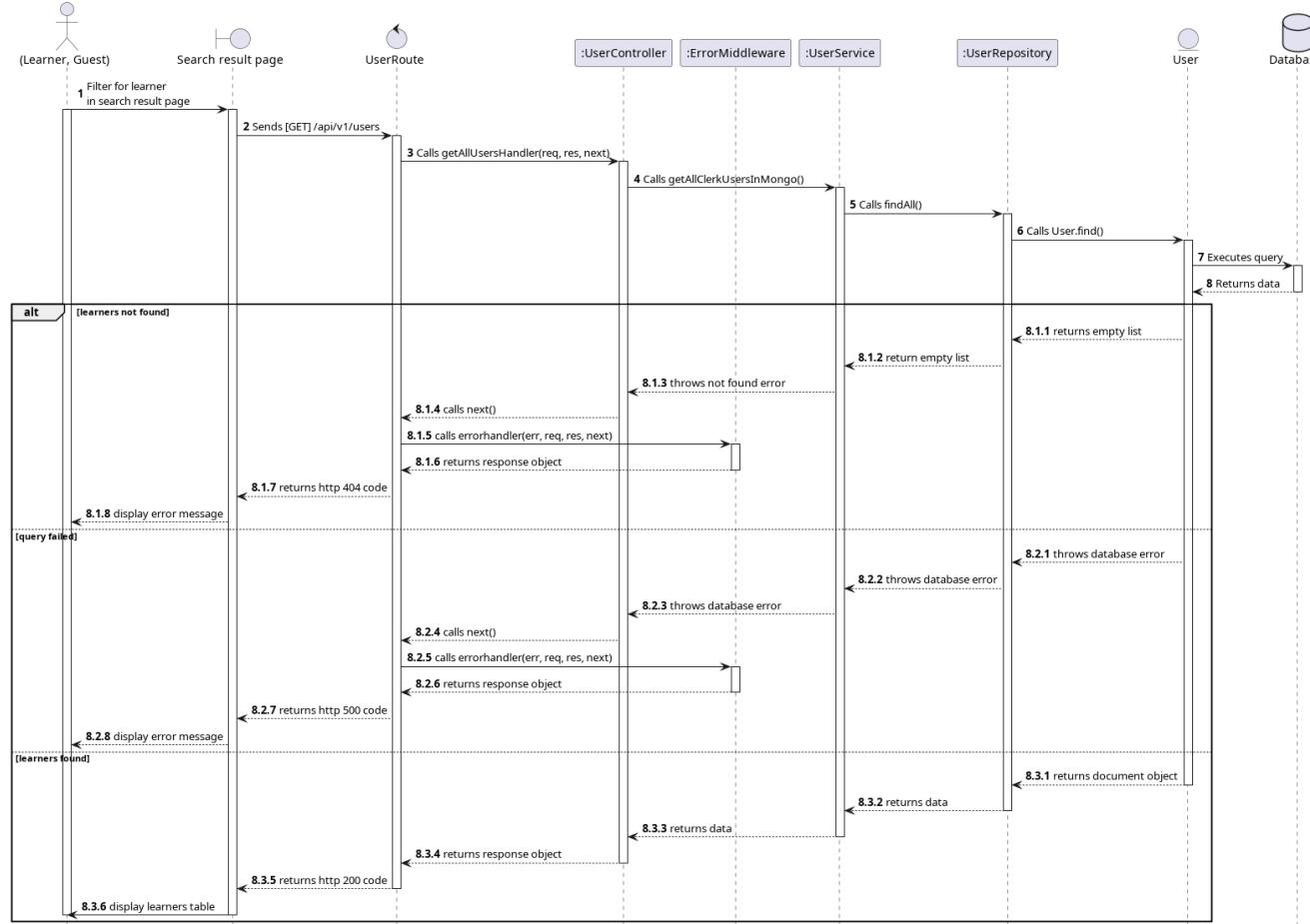


Figure 3.7.1.2.a: Learners and guests view learners¹³¹

131 [view learners.png](#)

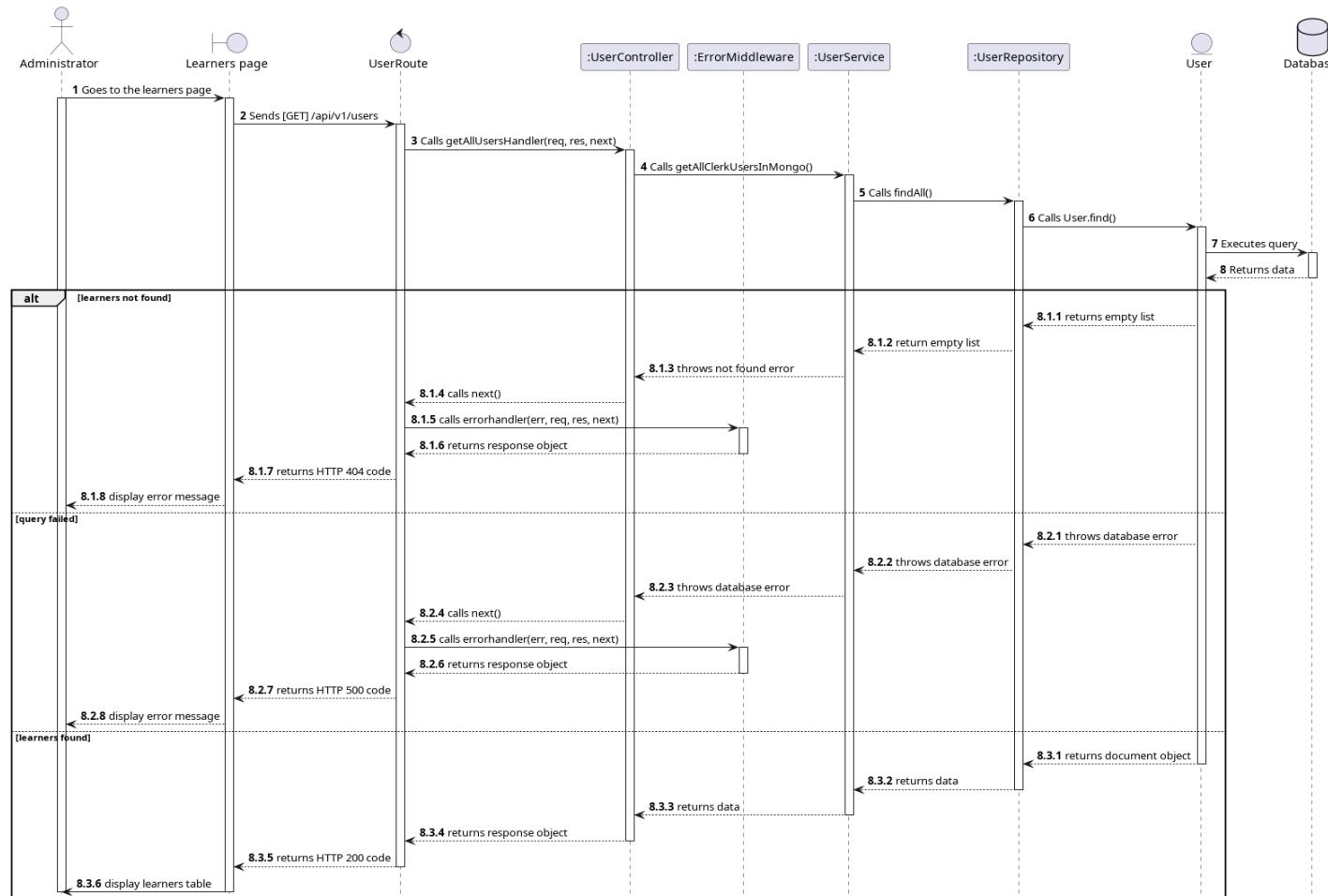


Figure 3.7.1.2.b: Administrator view learners¹³²

132 [view learners.png](#)

3.7.2 View learner profile

3.7.2.1 Class diagram

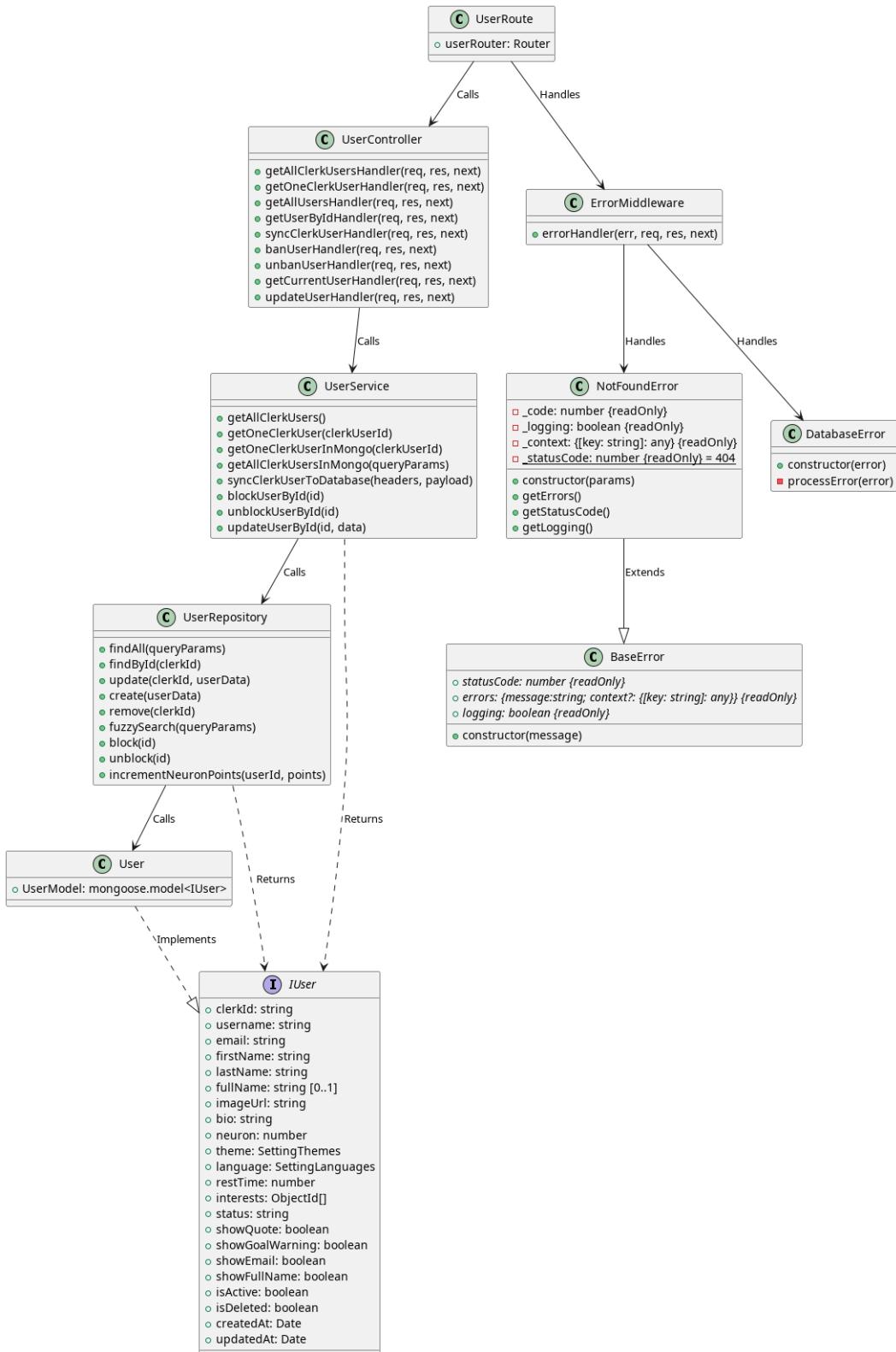


Figure 3.7.2.1: View learner profile¹³³

¹³³ [view learner detail.png](#)

3.7.2.2 Sequence diagram

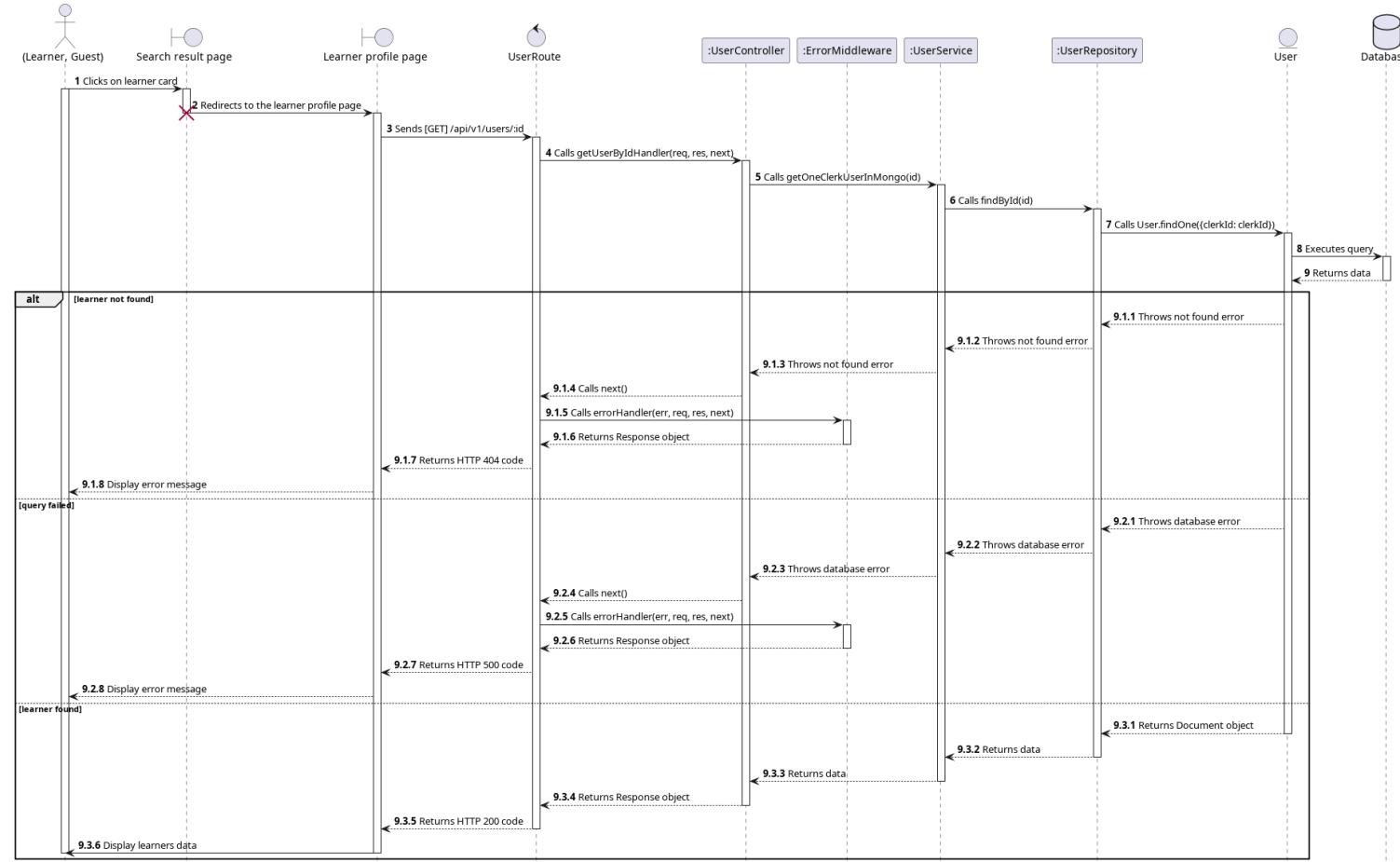


Figure 3.7.2.2.a: Learners and guests view learner profile¹³⁴

¹³⁴ [view learner detail.png](#)

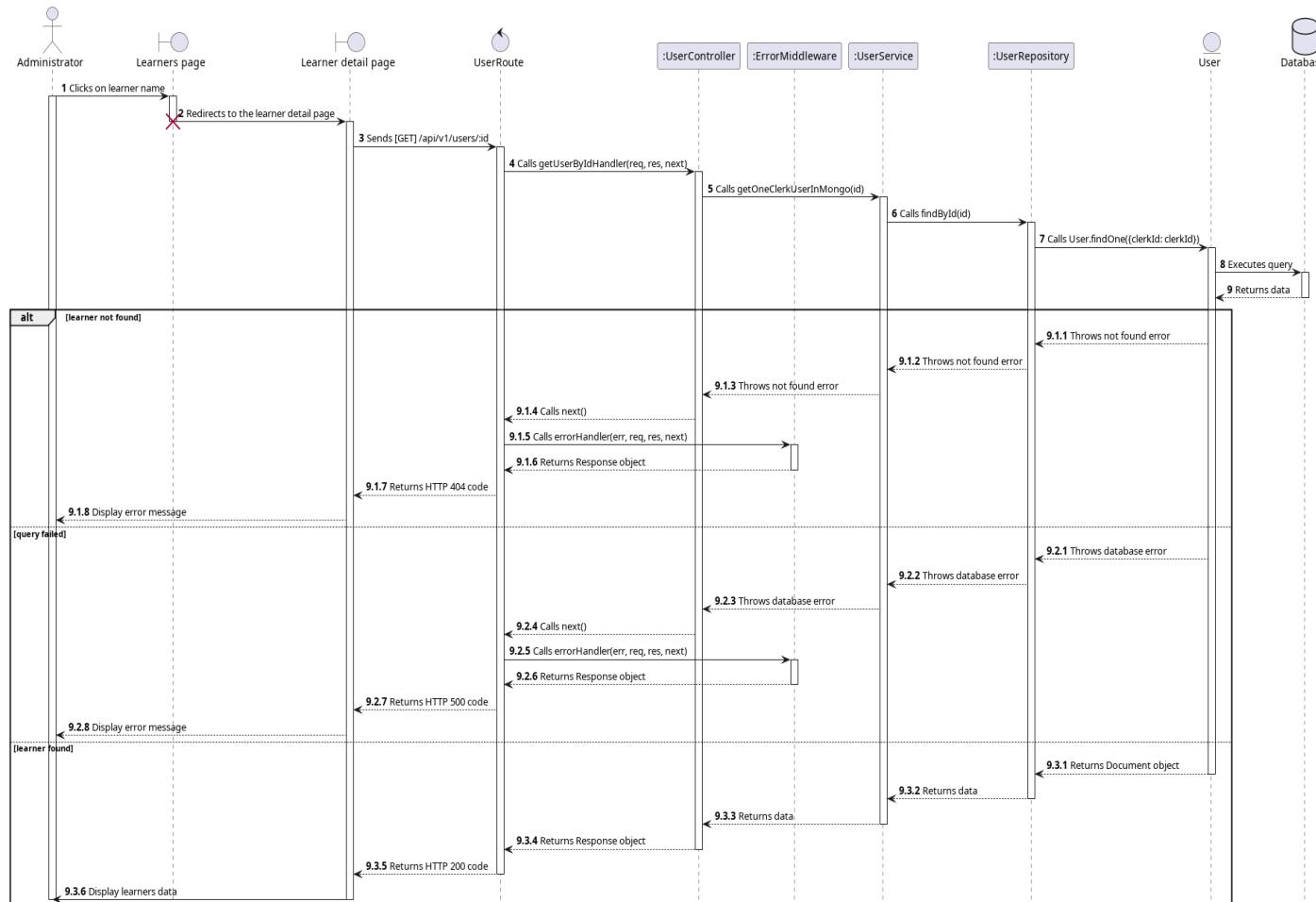


Figure 3.7.2.2.b: Administrator view learner profile¹³⁵

135 [view learner detail.png](#)

3.7.3 Update learner accessibilities

3.7.3.1 Class diagram

3.7.3.1.1 Block learner

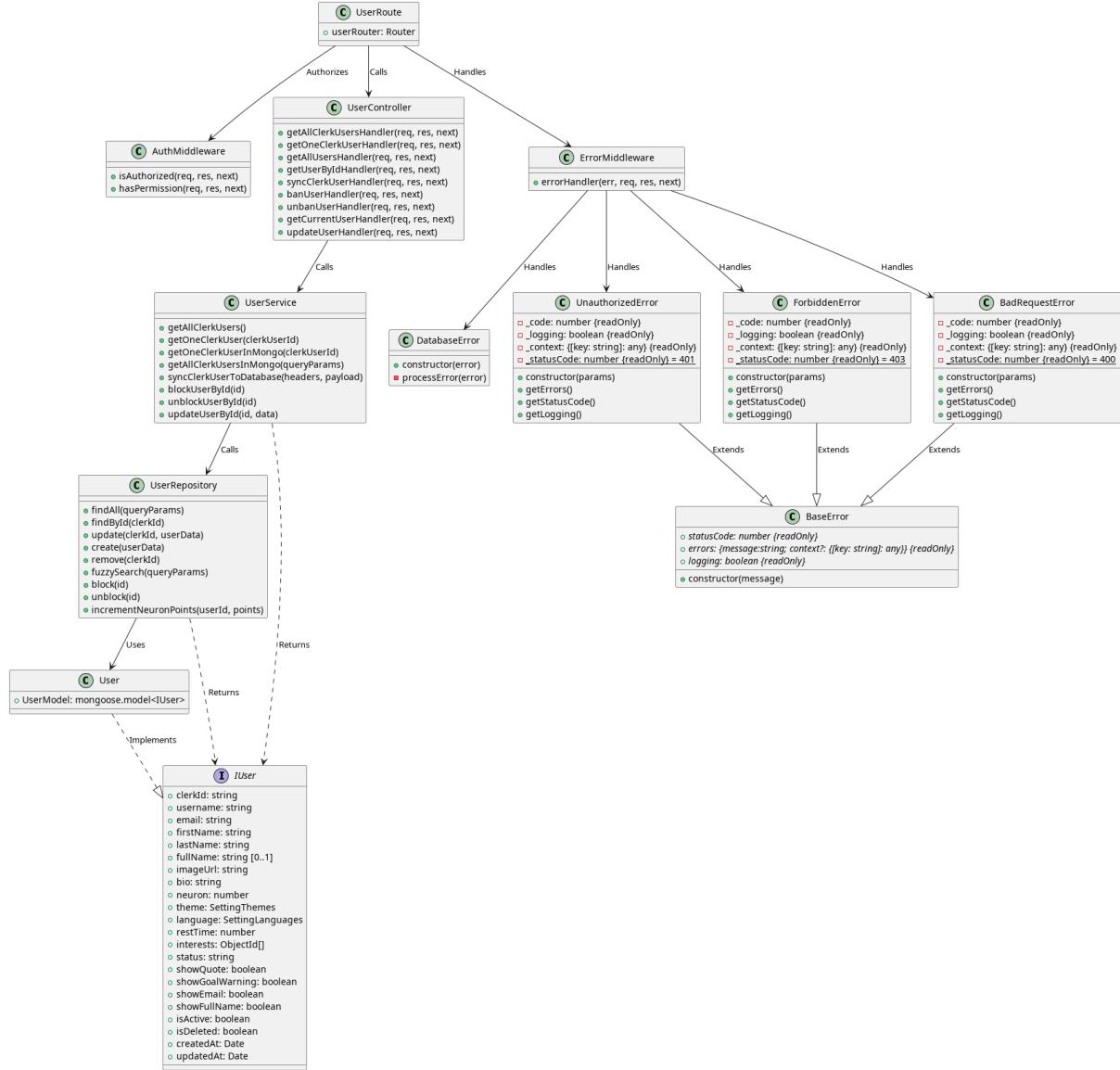


Figure 3.7.1.1: Block learner¹³⁶

136 [block learner.png](#)

3.7.3.1.2 Unblock learner

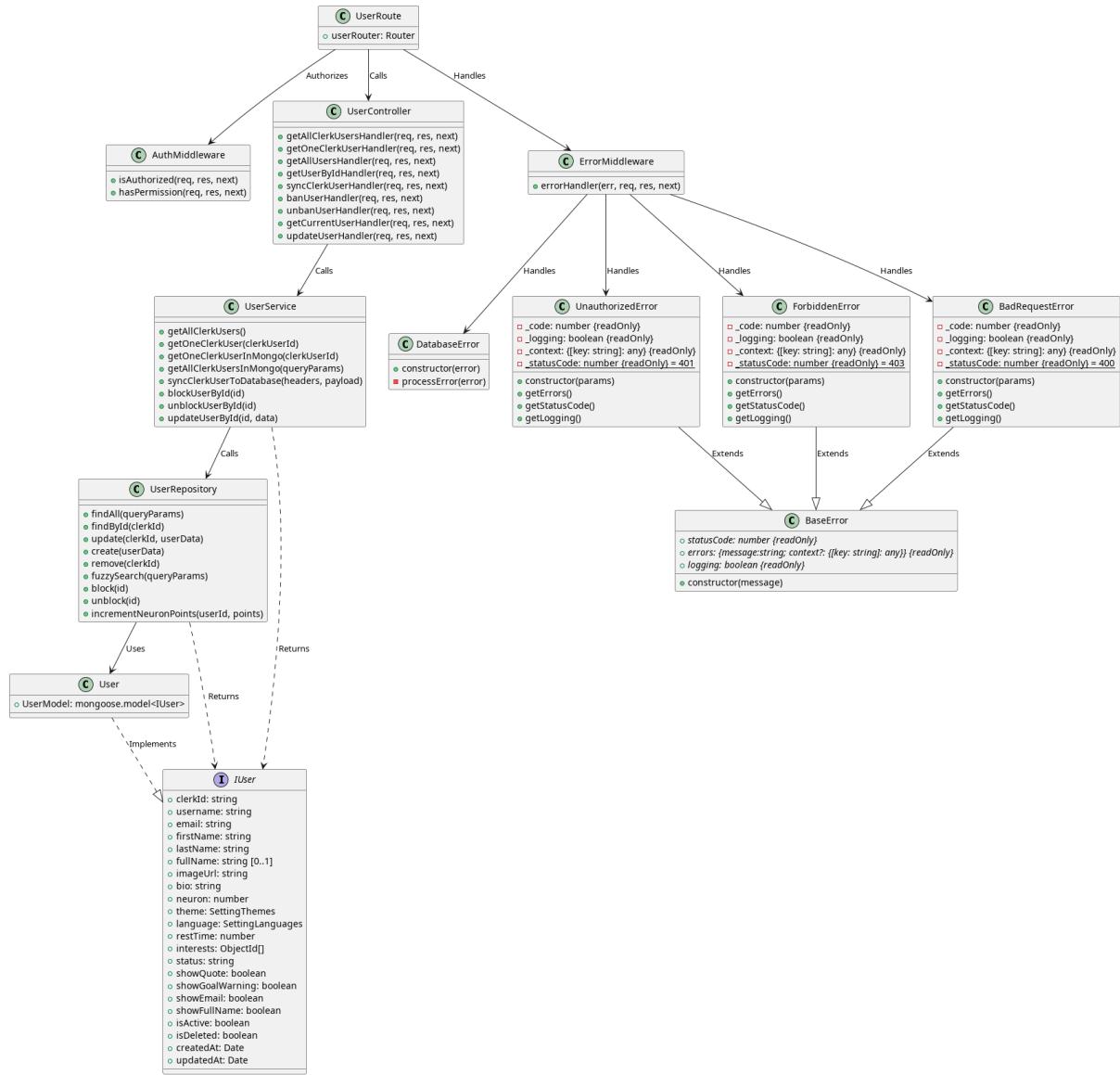


Figure 3.7.3.1.2: Unblock learner¹³⁷

¹³⁷ [unblock learner.png](#)

3.7.3.2 Sequence diagram

3.7.3.2.1 Block learner

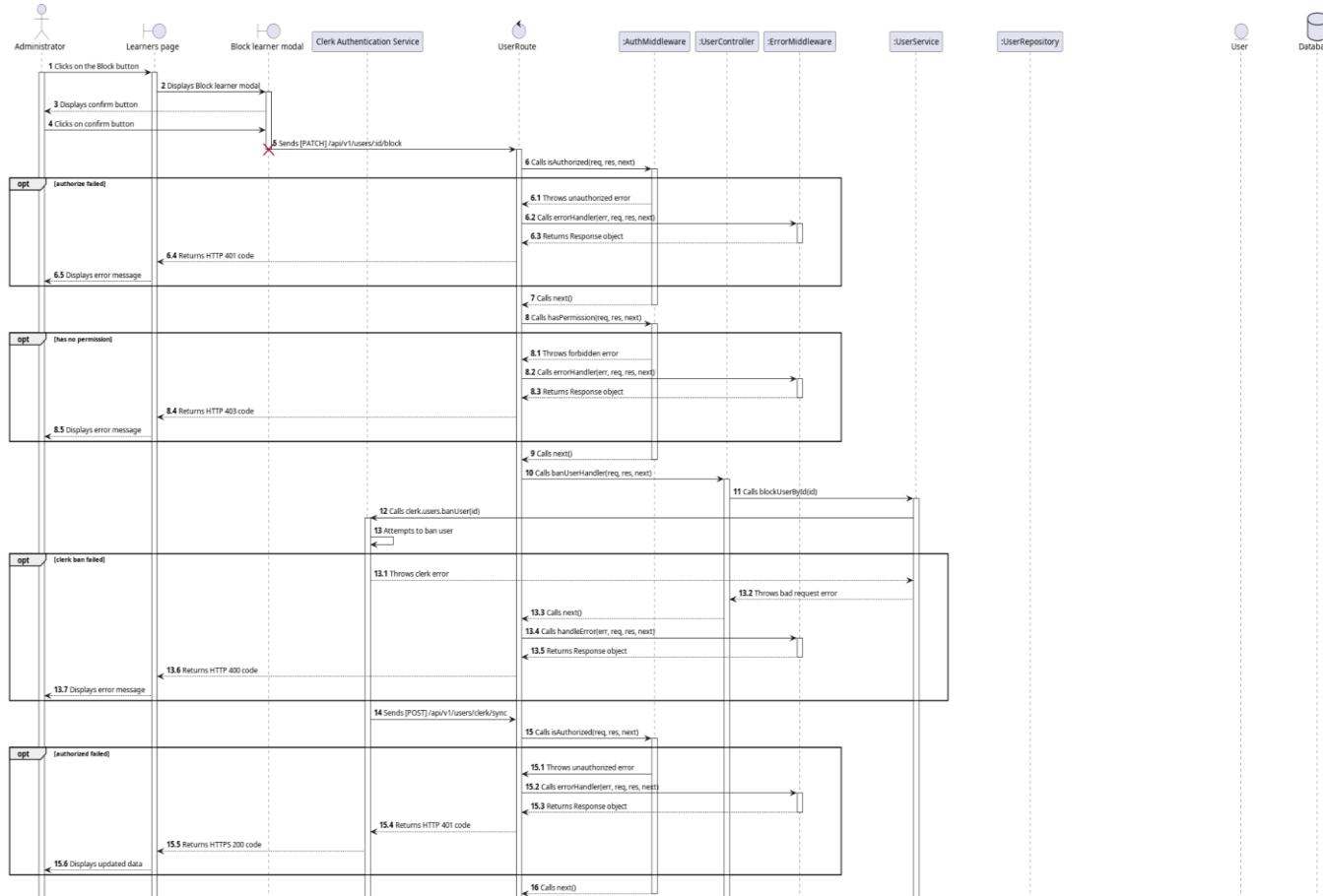


Figure 3.7.3.2.1.a: Block learner - 1¹³⁸

138 [block learner.png](#)

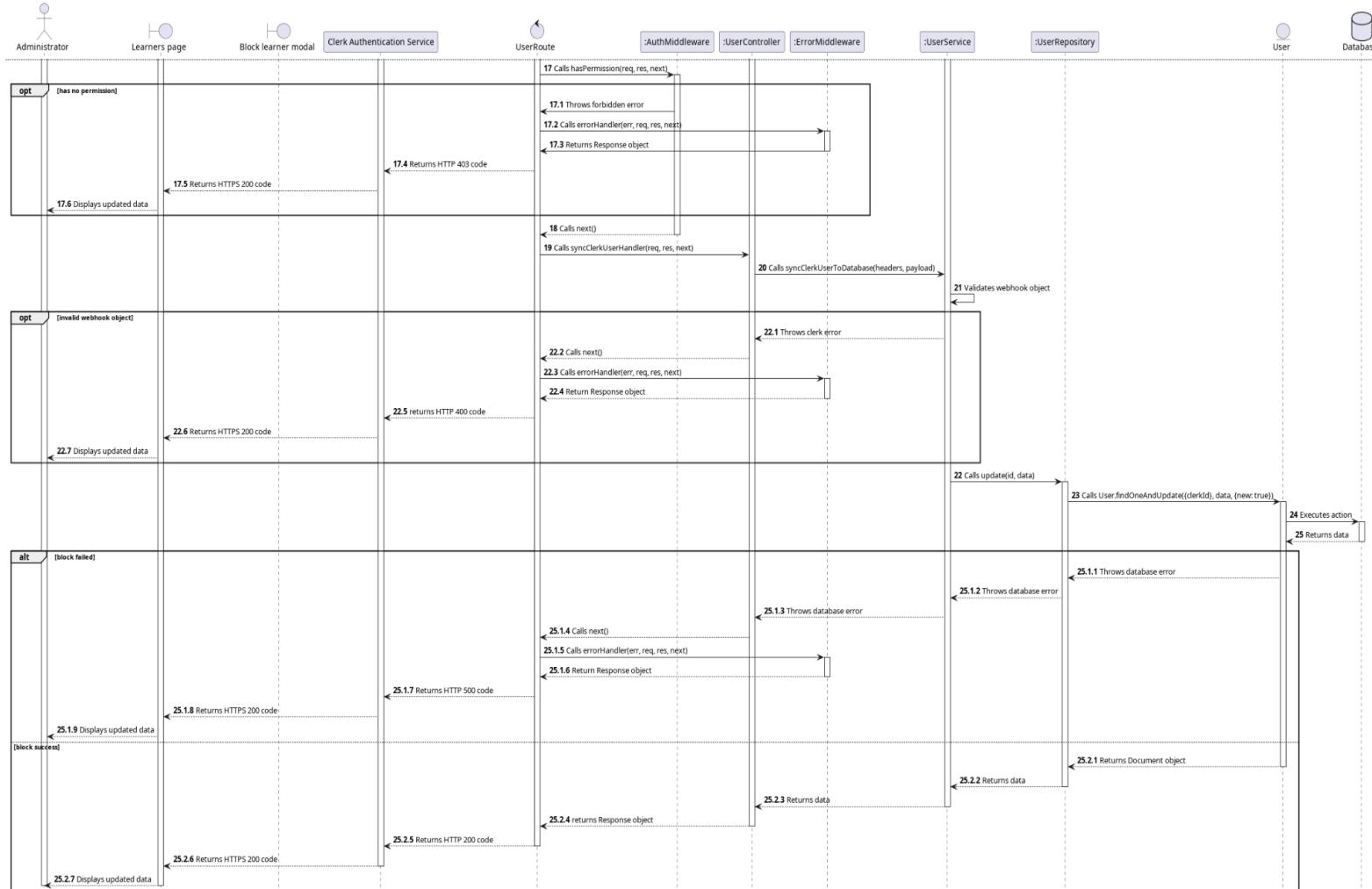


Figure 3.7.3.2.1.b: Block learner - 2

3.7.3.2.2 Unblock learner

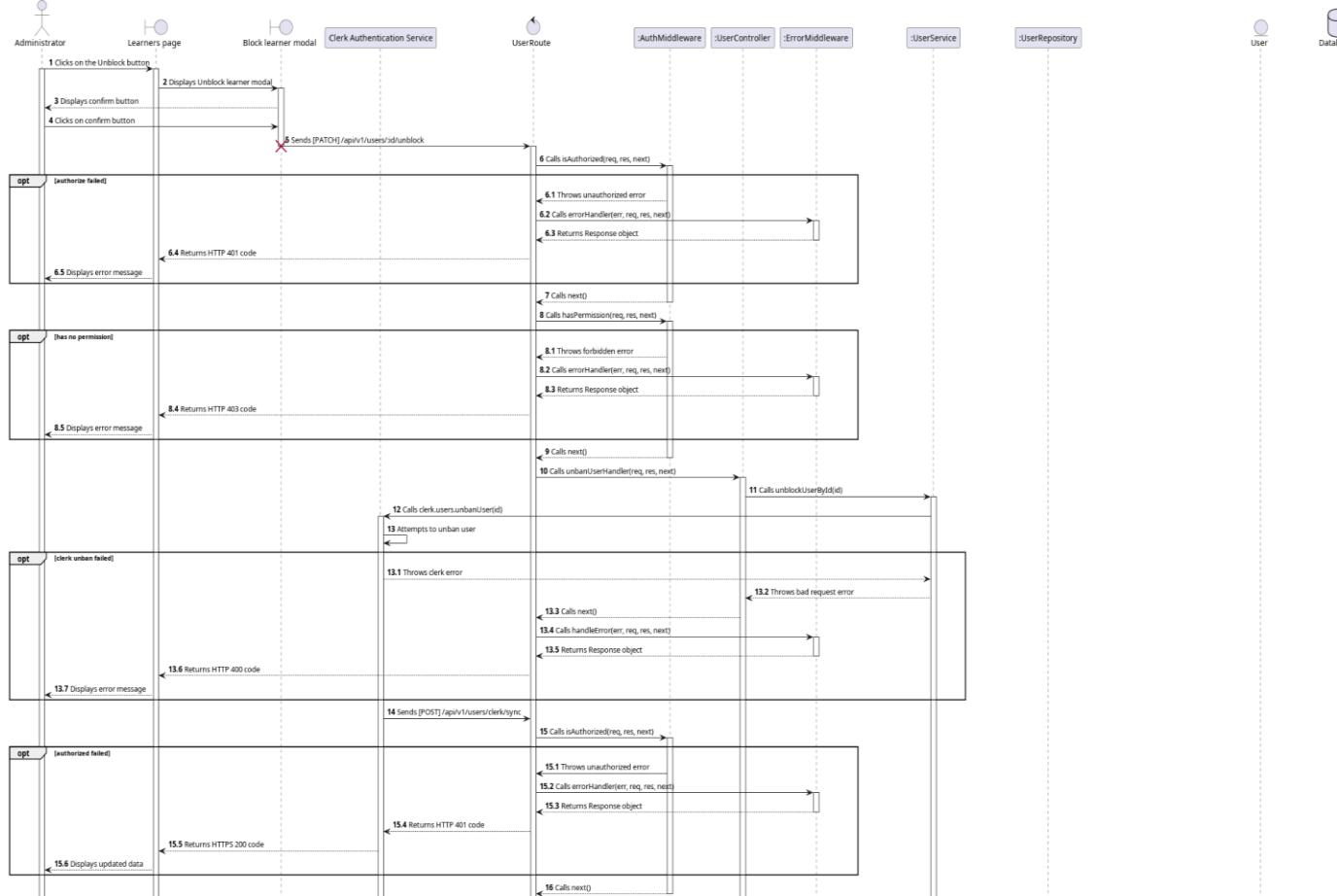


Figure 3.7.3.2.2.a: Unblock learner - 1¹³⁹

139 [unblock.png](#)

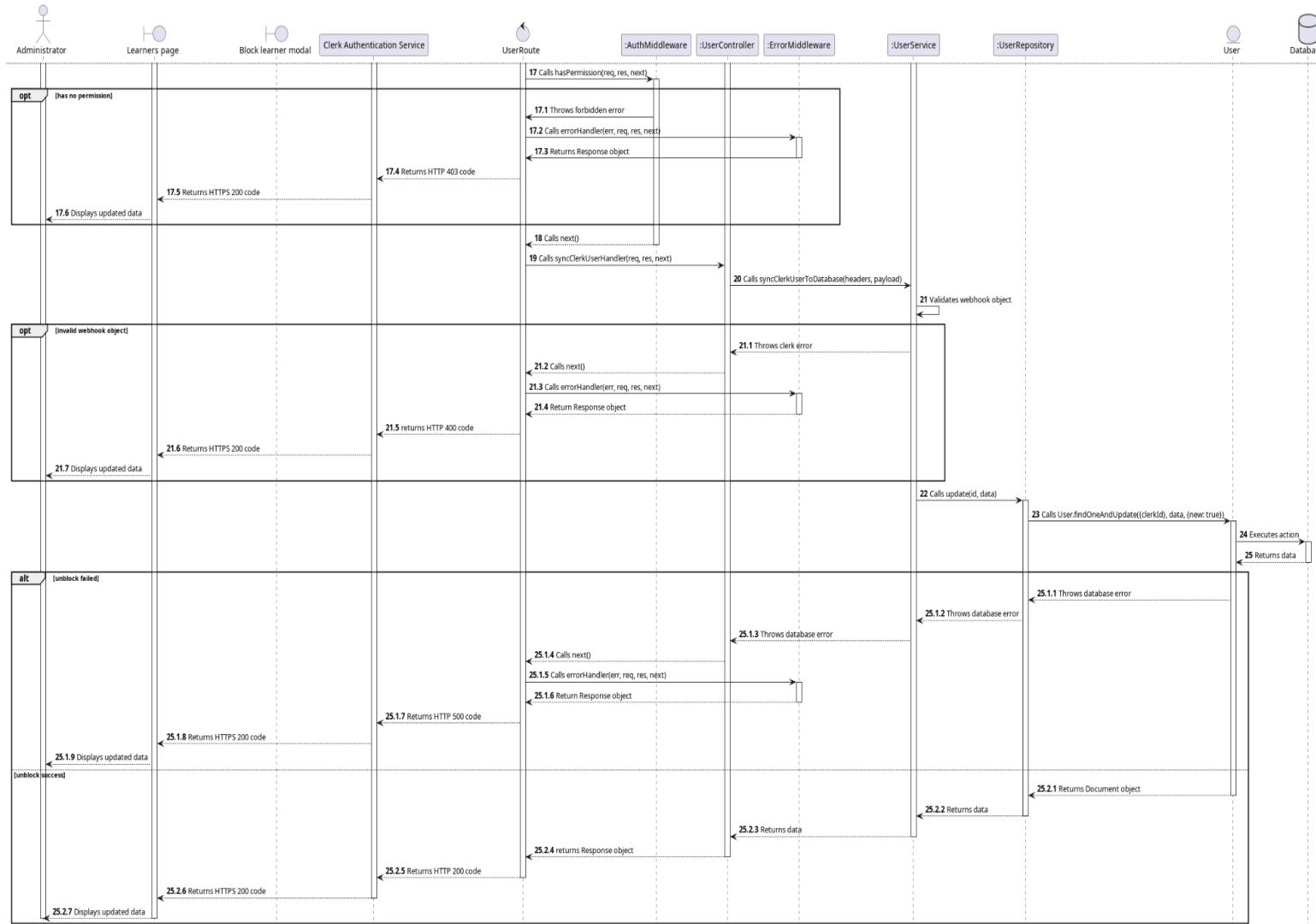


Figure 3.7.3.2.2.b: Unblock learner - 2

3.8 Report Management

3.8.1 View reports

3.8.1.1 Class diagram

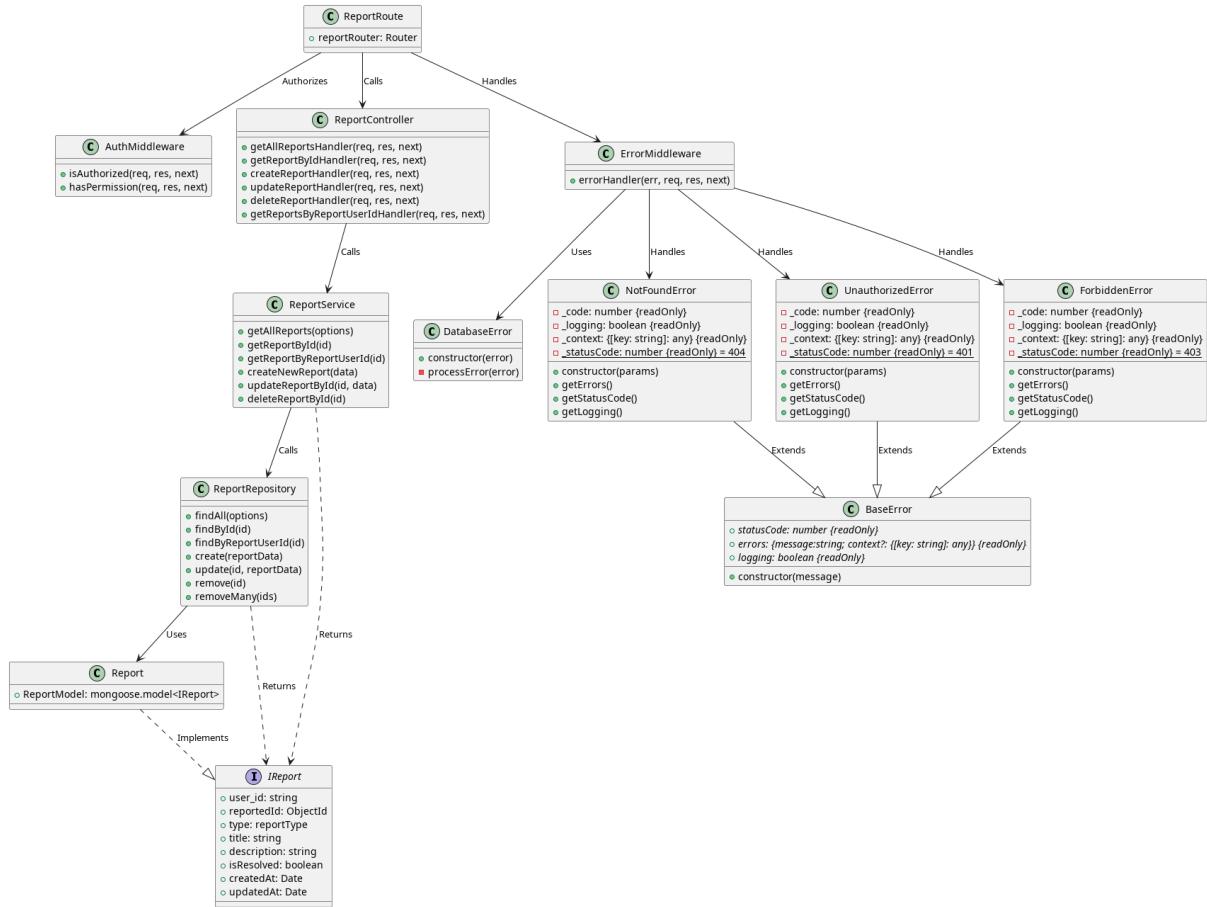


Figure 3.8.1.1: View reports¹⁴⁰

140 [view reports.png](#)

3.8.1.2 Sequence diagram

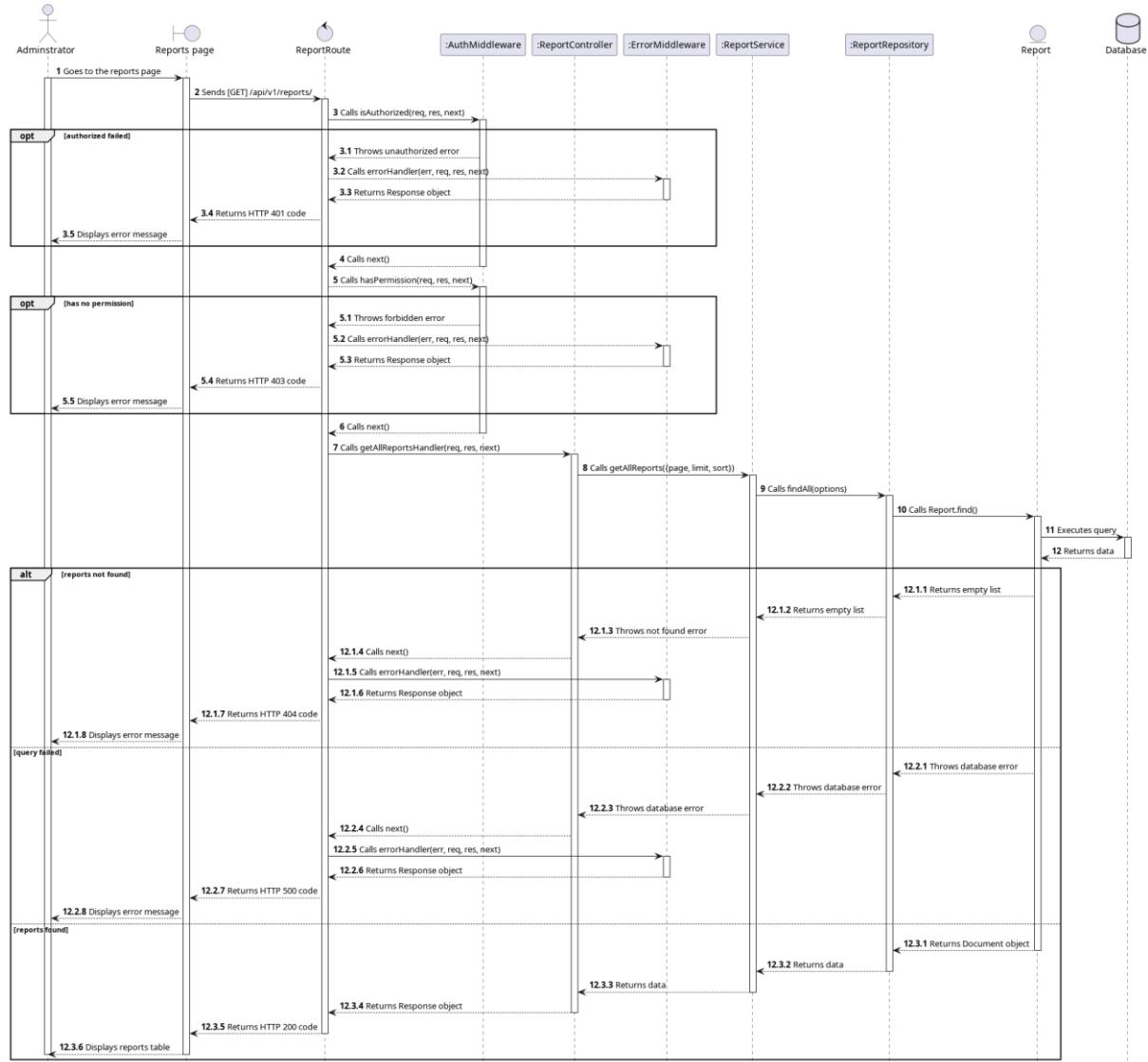


Figure 3.8.1.2: View reports¹⁴¹

141 [view reports.png](#)

3.8.2 View report detail

3.8.2.1 Class diagram

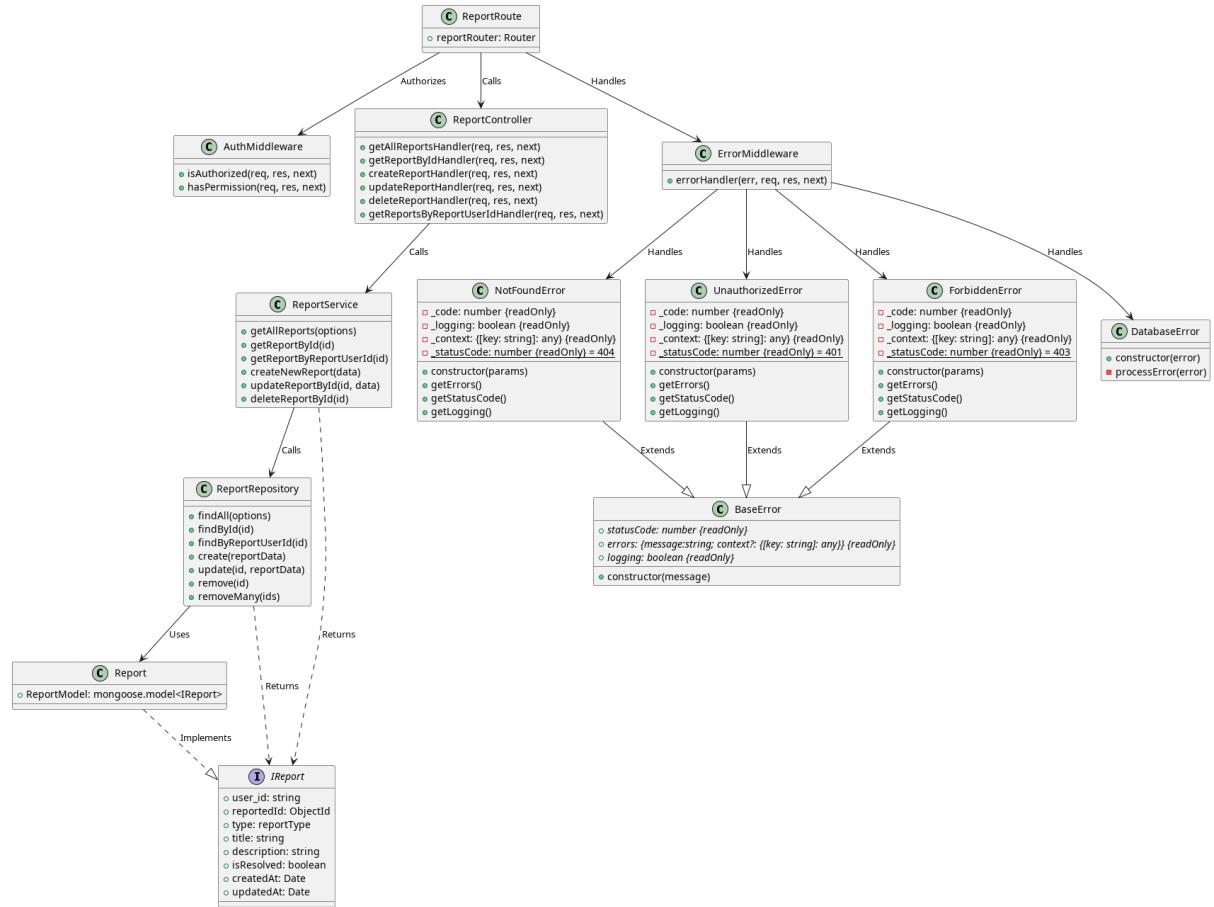


Figure 3.8.2.1: View report detail¹⁴²

¹⁴² [view report detail.png](#)

3.8.2.2 Sequence diagram

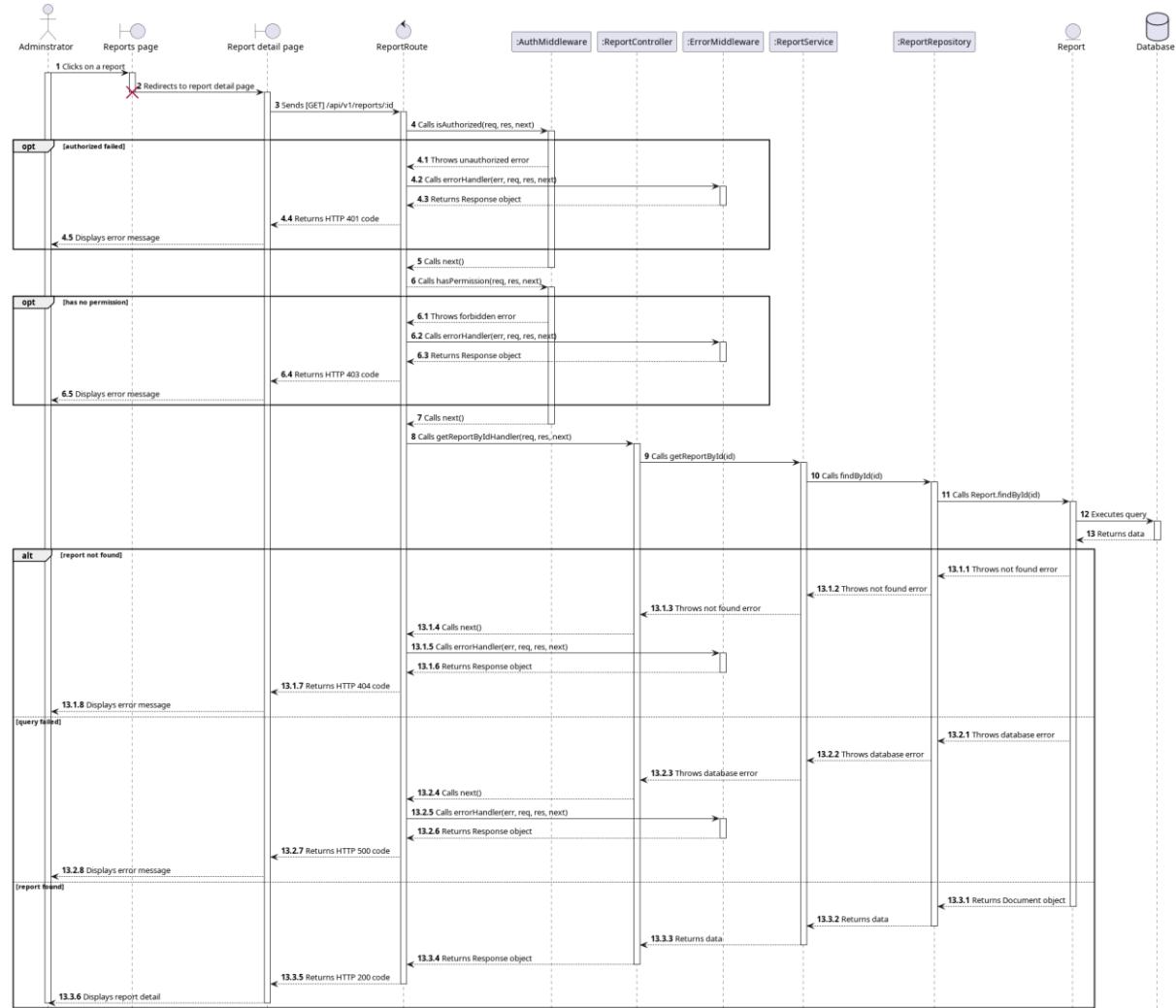


Figure 3.8.2.2: View report detail¹⁴³

¹⁴³ [view report detail.png](#)

3.8.3 Create report

3.8.3.1 Class diagram

3.8.3.1.1 Report question set

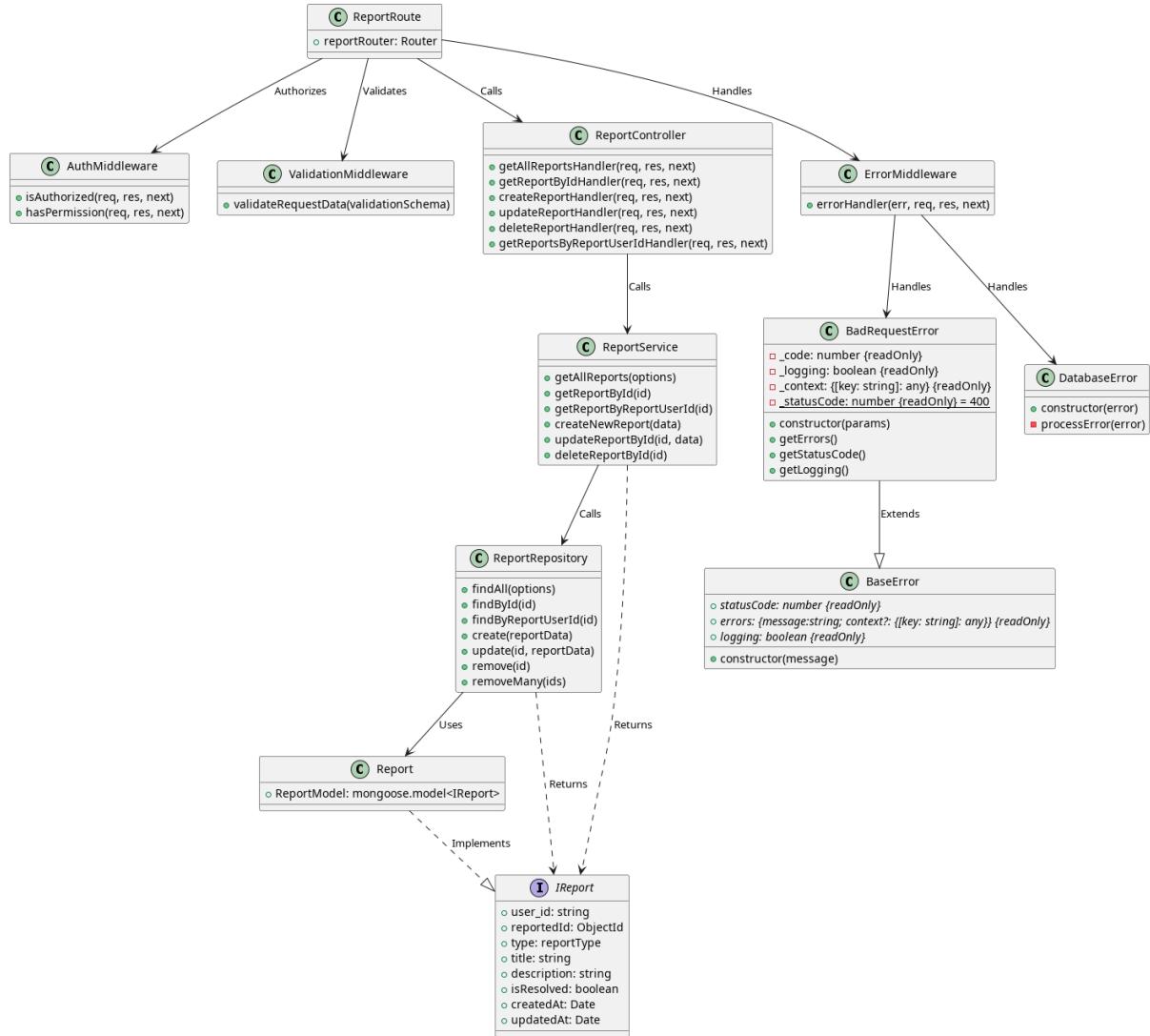


Figure 3.8.3.1.1: Report question set¹⁴⁴

¹⁴⁴ [report question set.png](#)

3.8.3.1.2 Report comment

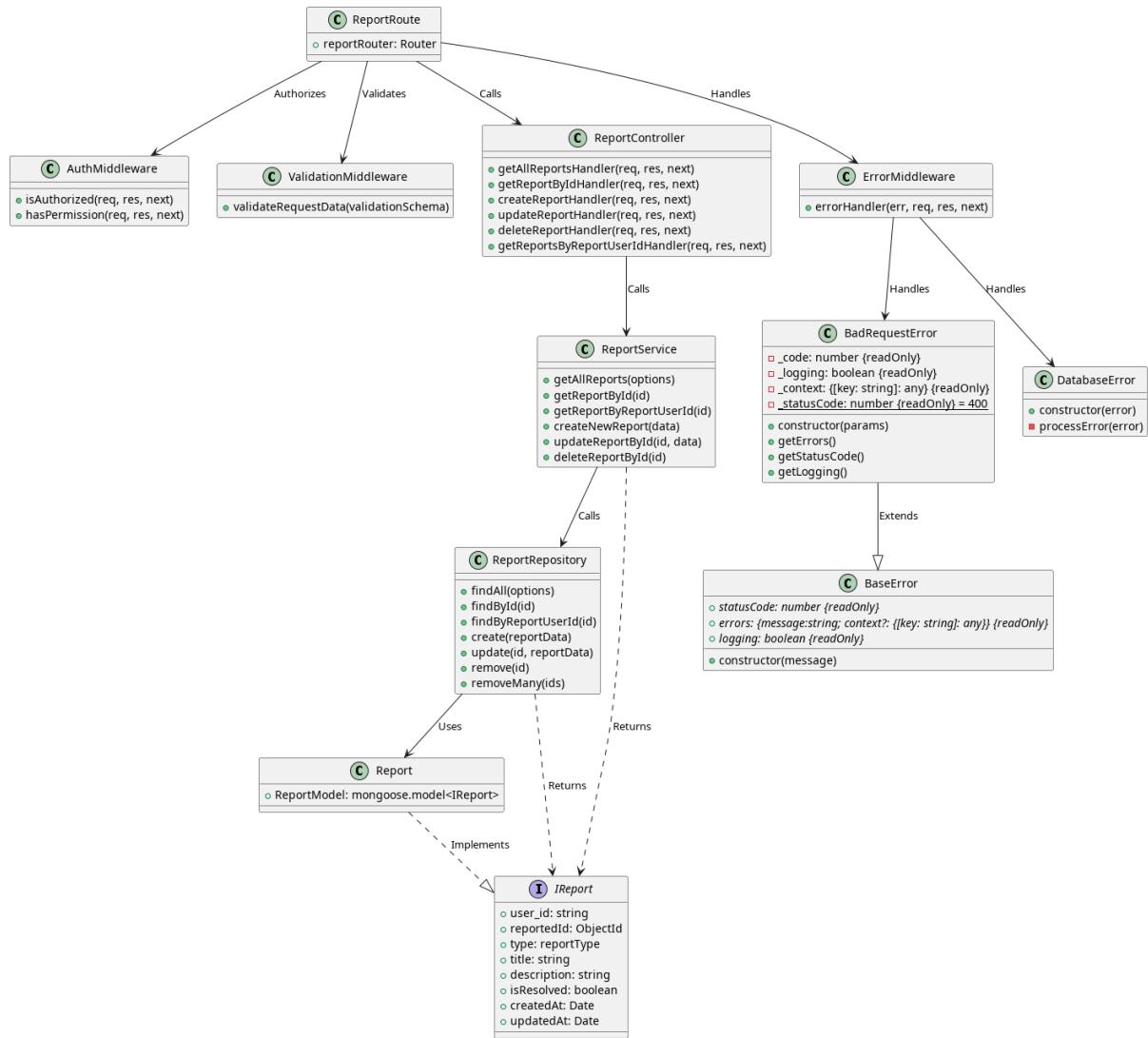


Figure 3.8.3.1.2: Report comment¹⁴⁵

¹⁴⁵ [report comment.png](#)

3.8.3.2 Sequence diagram

3.8.3.2.1 Report question set

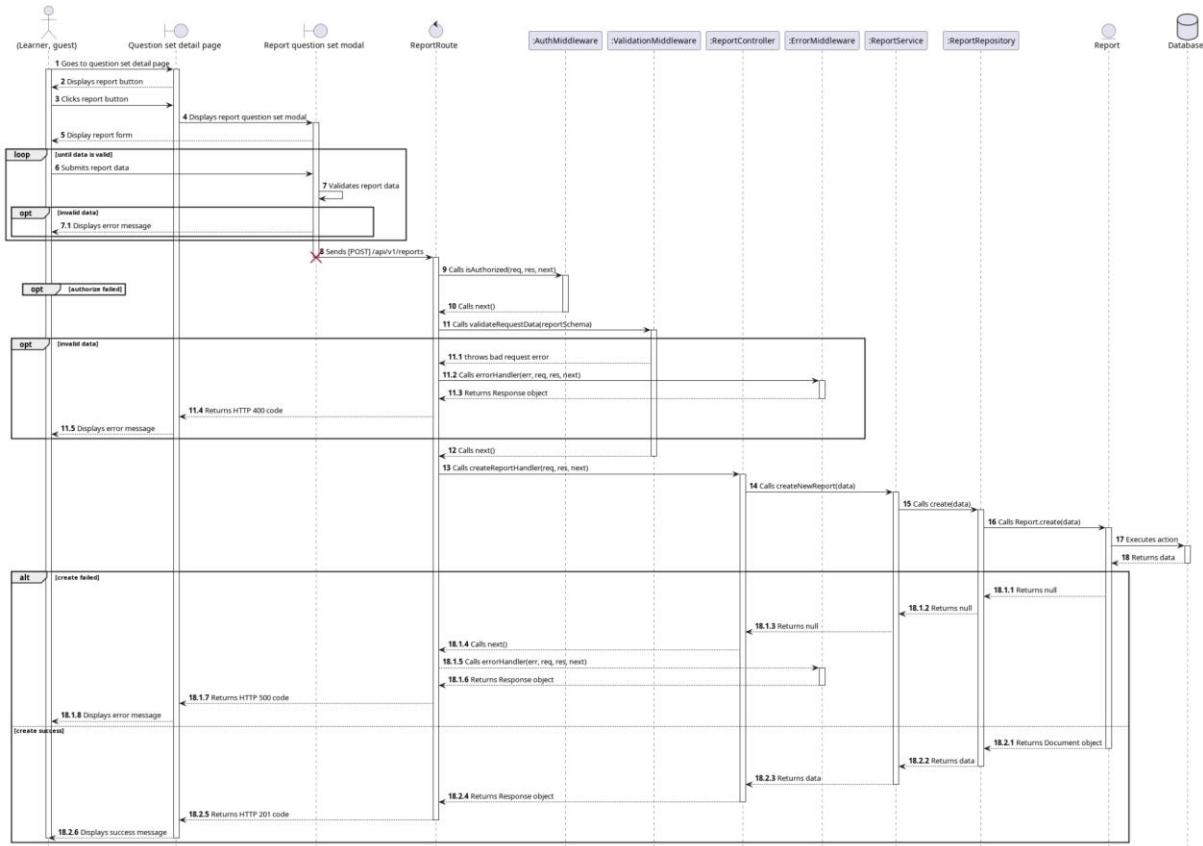


Figure 3.8.3.2.1: Report question set¹⁴⁶

146 [report question set.png](#)

3.8.3.2.2 Report comment

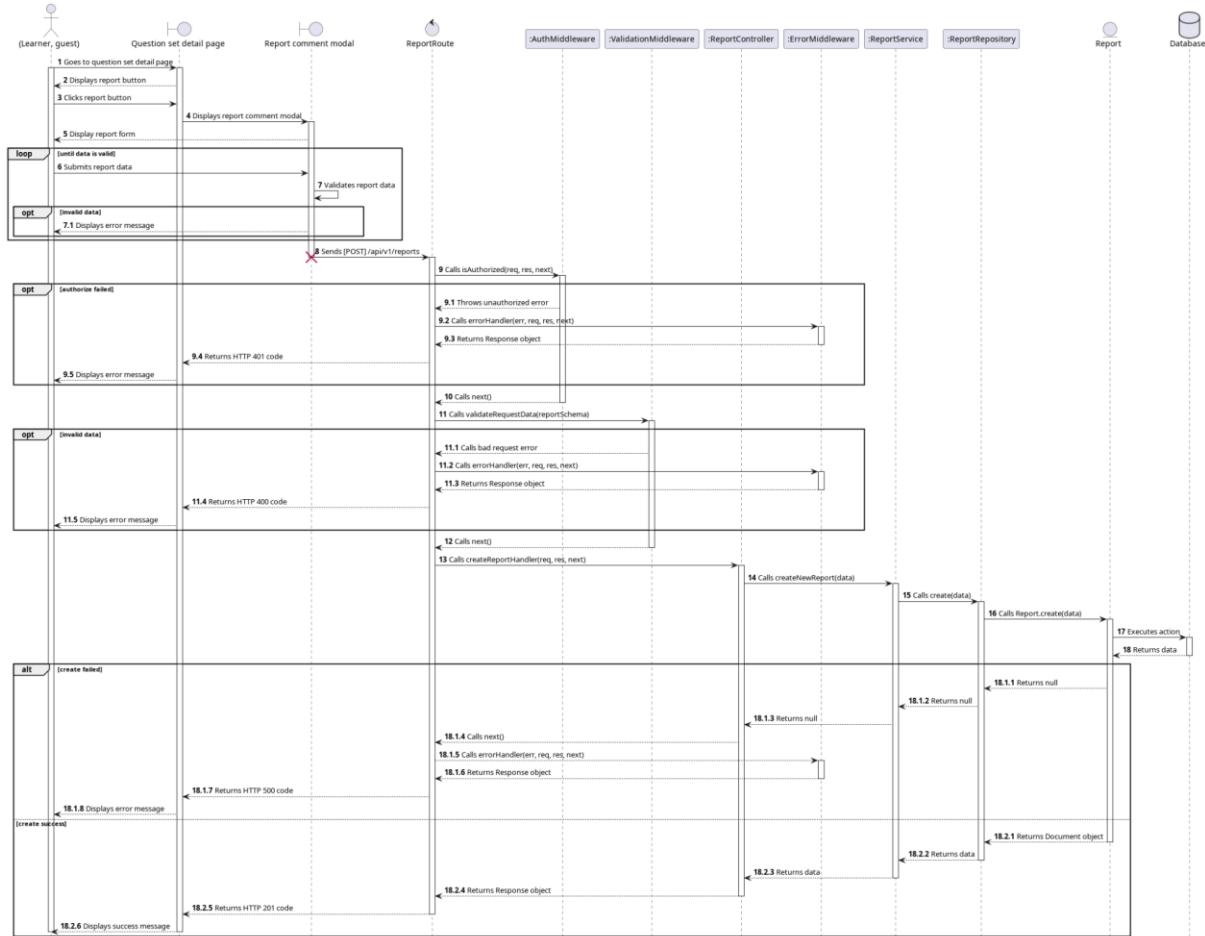


Figure 3.8.3.2.2: Report comment¹⁴⁷

¹⁴⁷ report comment.png

3.8.4 Update report statuses

3.8.4.1 Class diagram

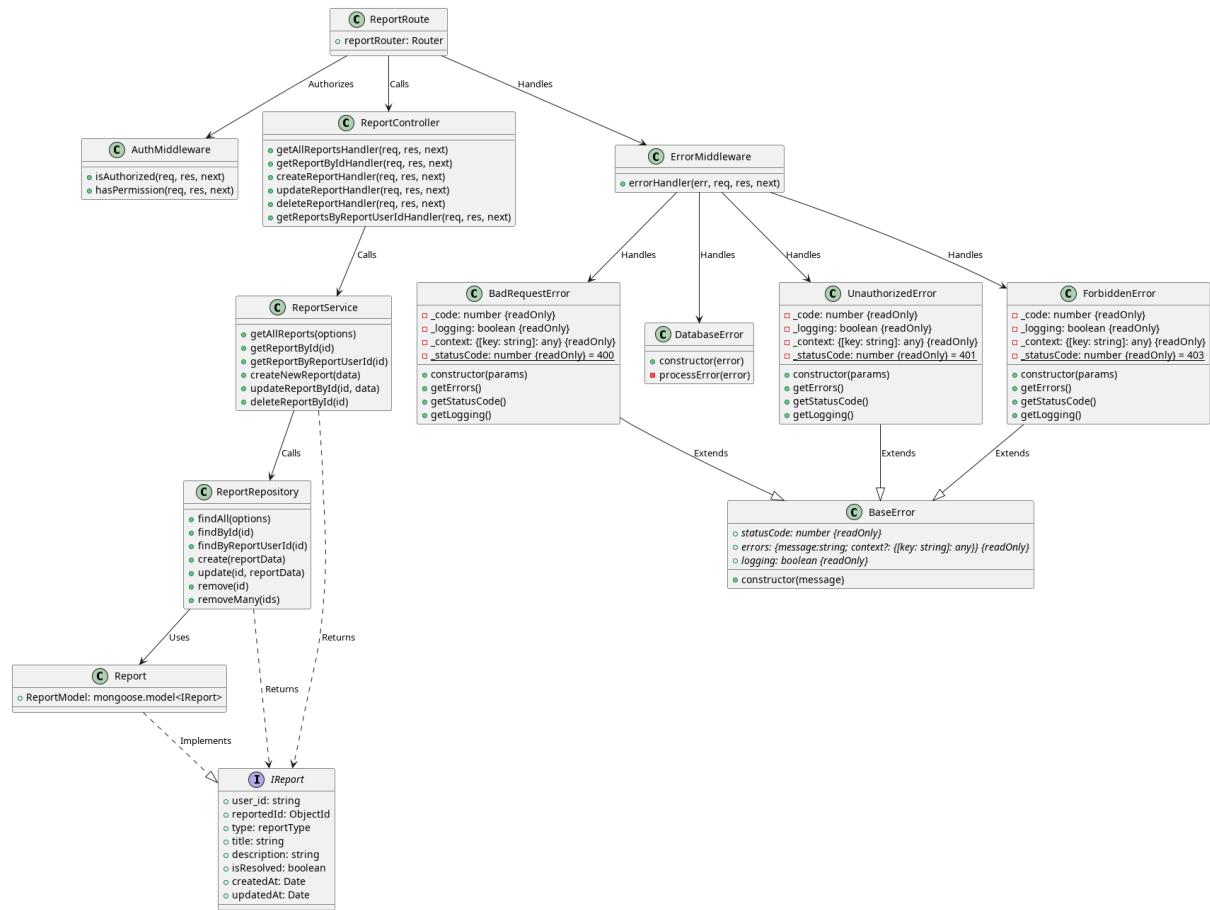


Figure 3.8.4.1: Update report statuses¹⁴⁸

148 update report status.png

3.8.4.2 Sequence diagram

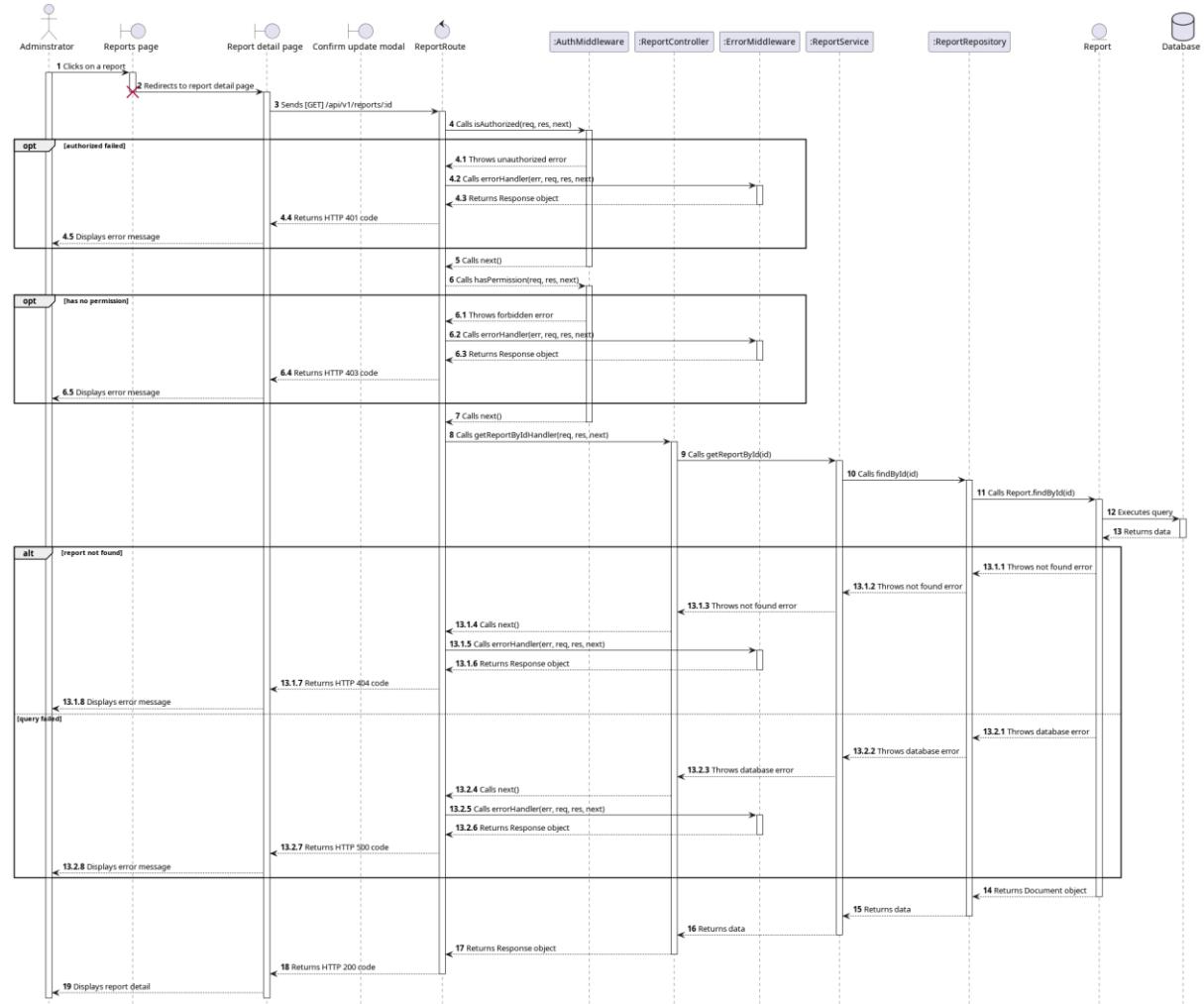


Figure 3.8.4.2.a: Get report detail¹⁴⁹

149 [view reports.png](#)

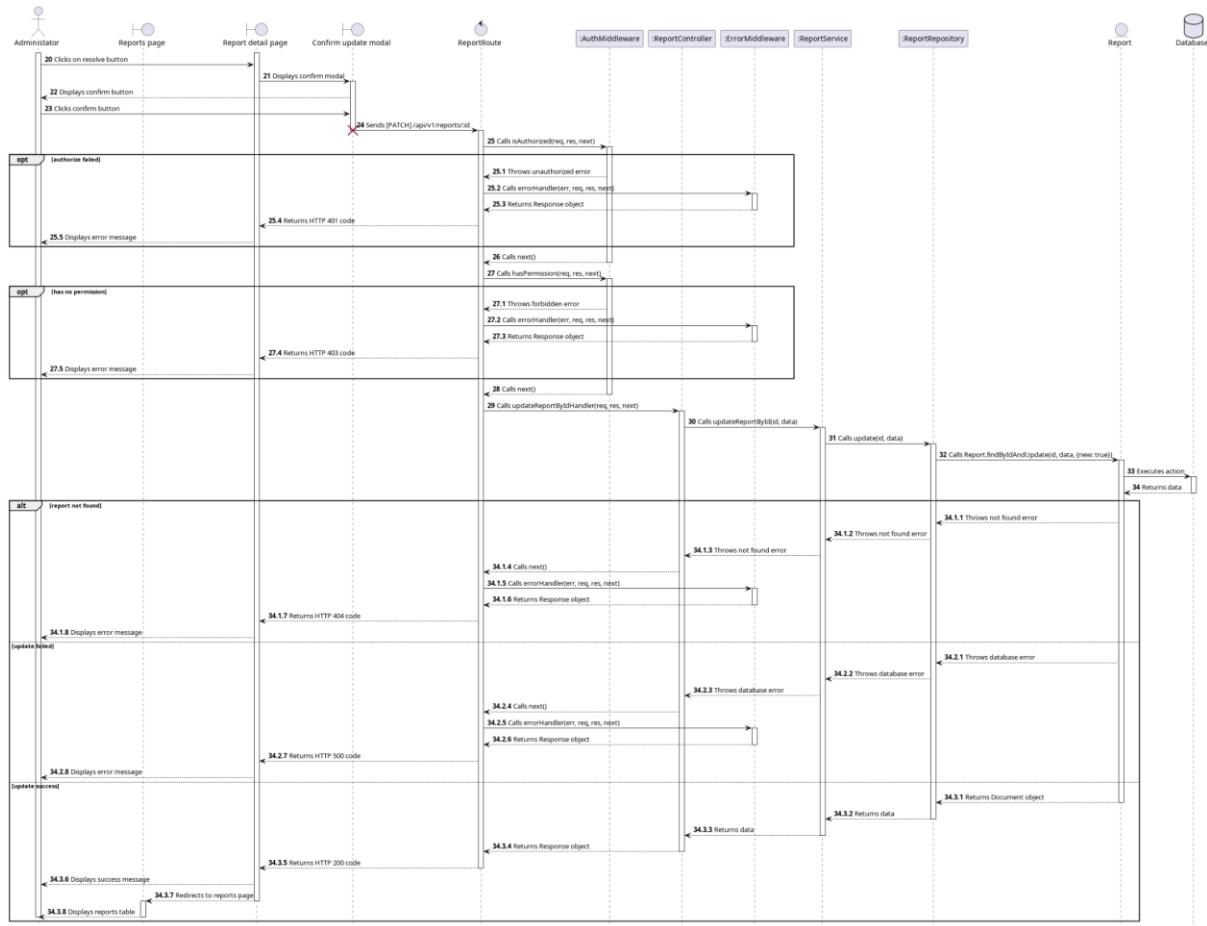


Figure 3.8.4.2.b: Update report statuses¹⁵⁰

150 [update report status.png](#)

3.9 Field Management

3.9.1 View fields

3.9.1.1 Class diagram

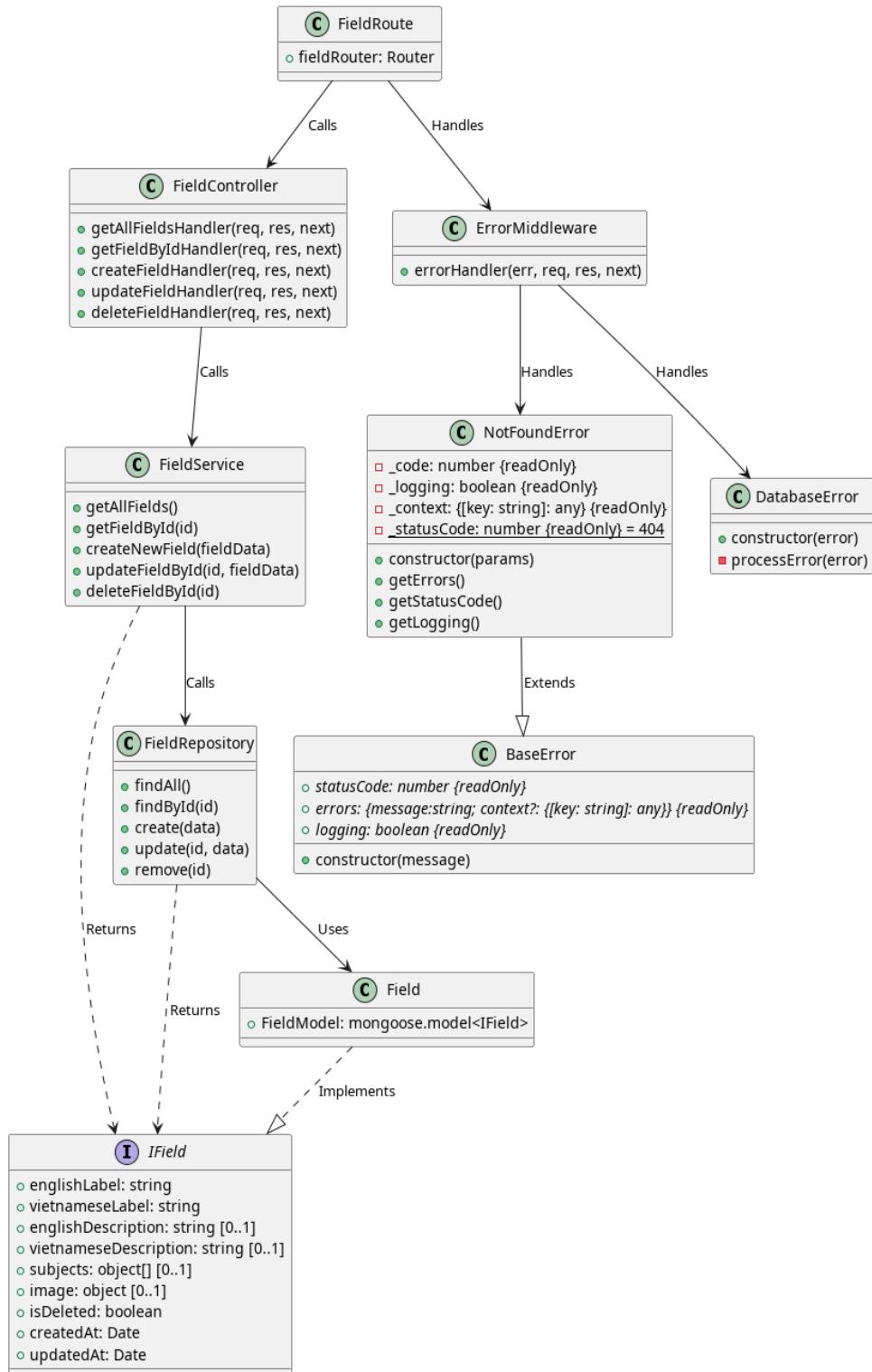


Figure 3.9.1.1: View fields¹⁵¹

¹⁵¹ [view all fields.png](#)

3.9.1.2 Sequence diagram

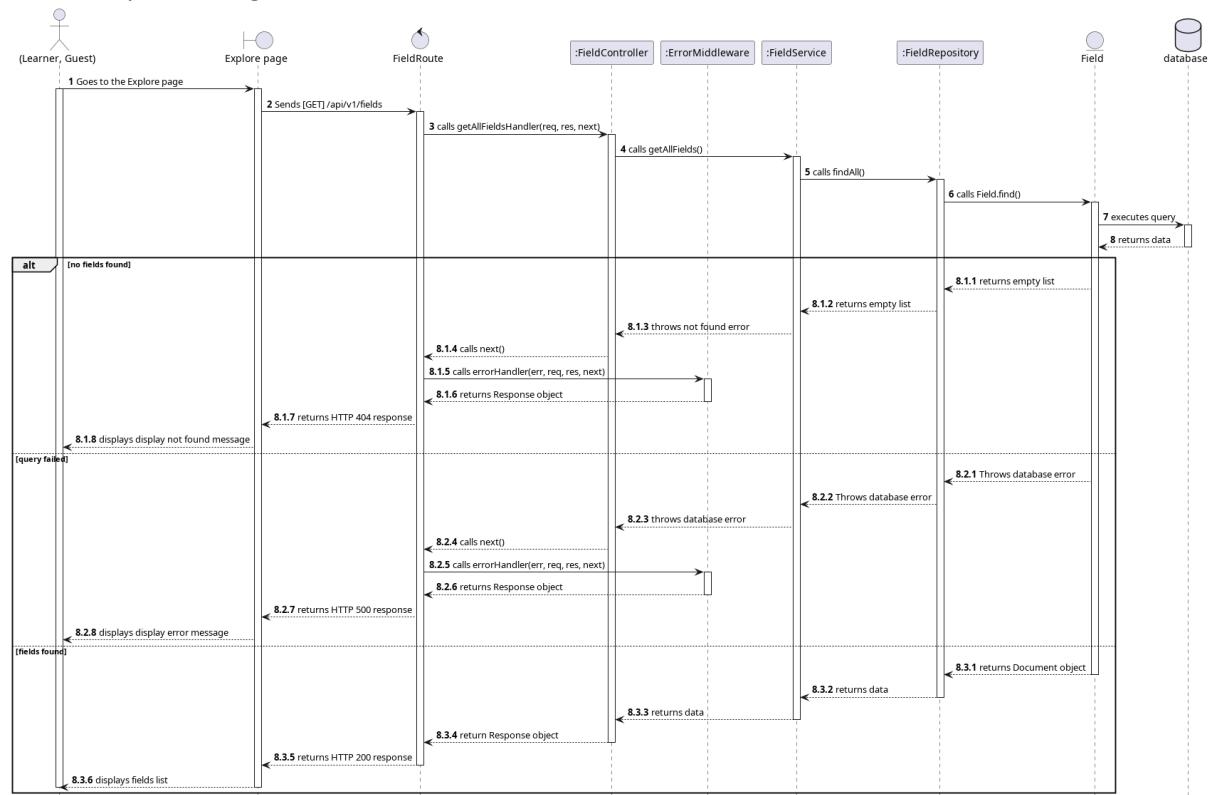


Figure 3.9.1.2.a: Learners and Guests view fields¹⁵²

152 [view fields.png](#)

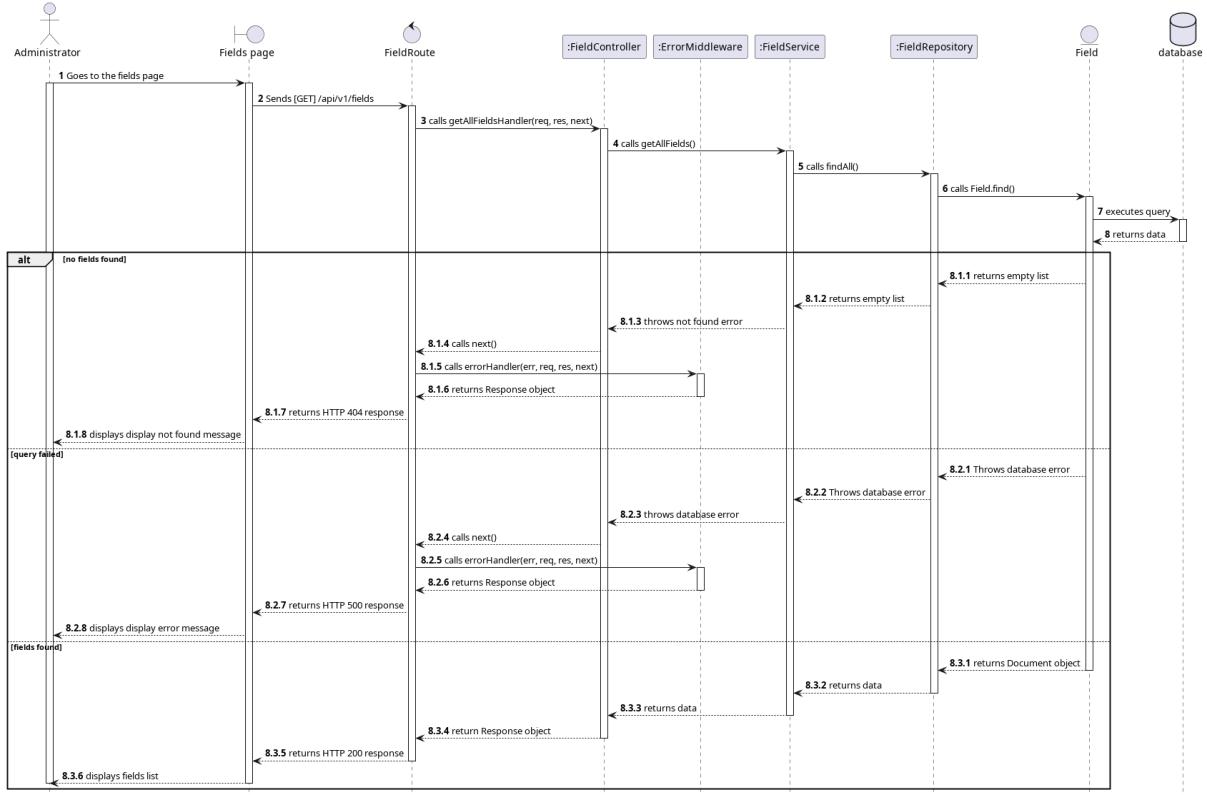


Figure 3.9.1.2.b: Administrators view fields¹⁵³

153 [view fields.png](#)

3.9.2 View field details

3.9.2.1 Class diagram

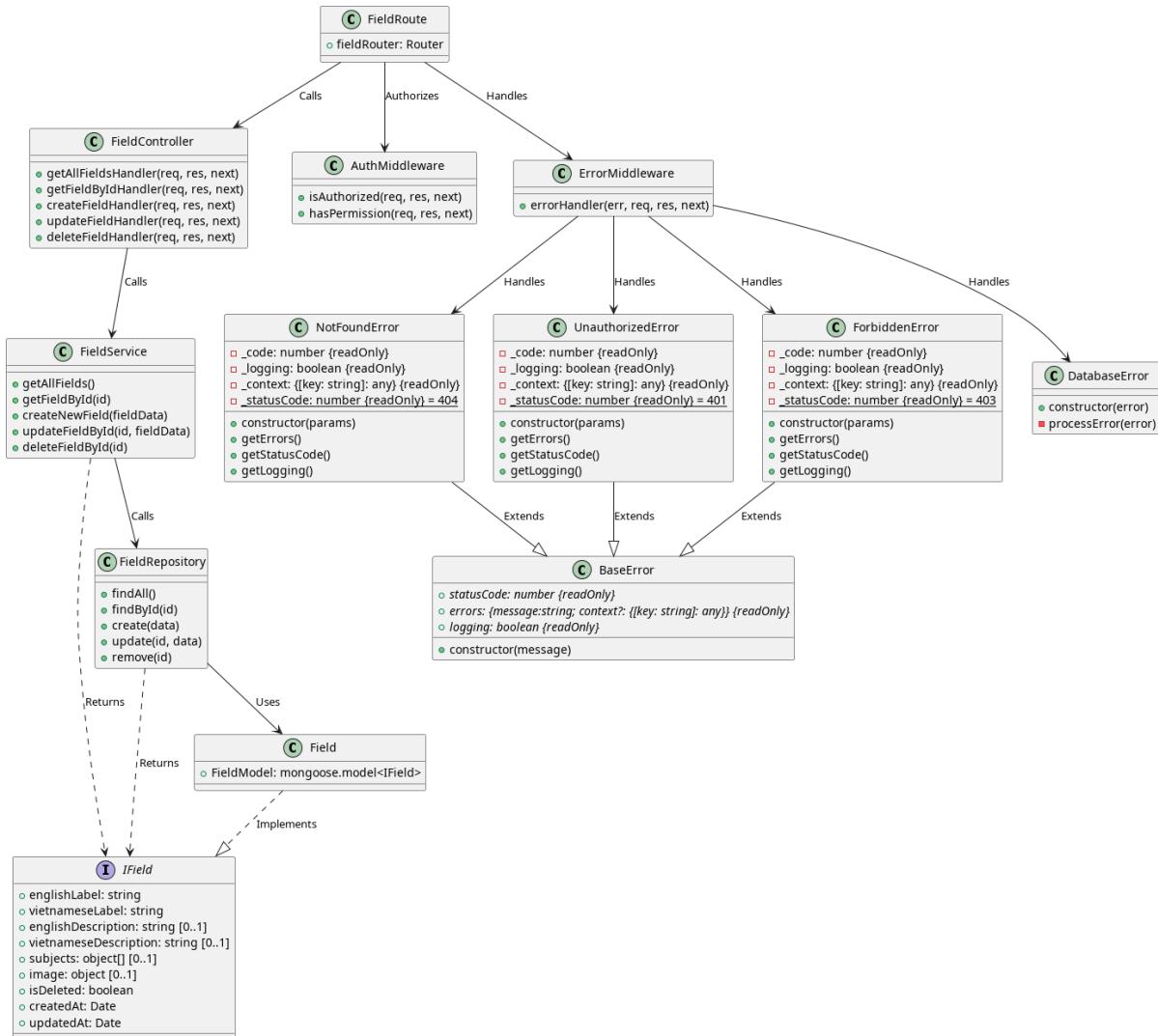


Figure 3.9.2.1: View field detail¹⁵⁴

154 [view field detail.png](#)

3.9.2.2 Sequence diagram

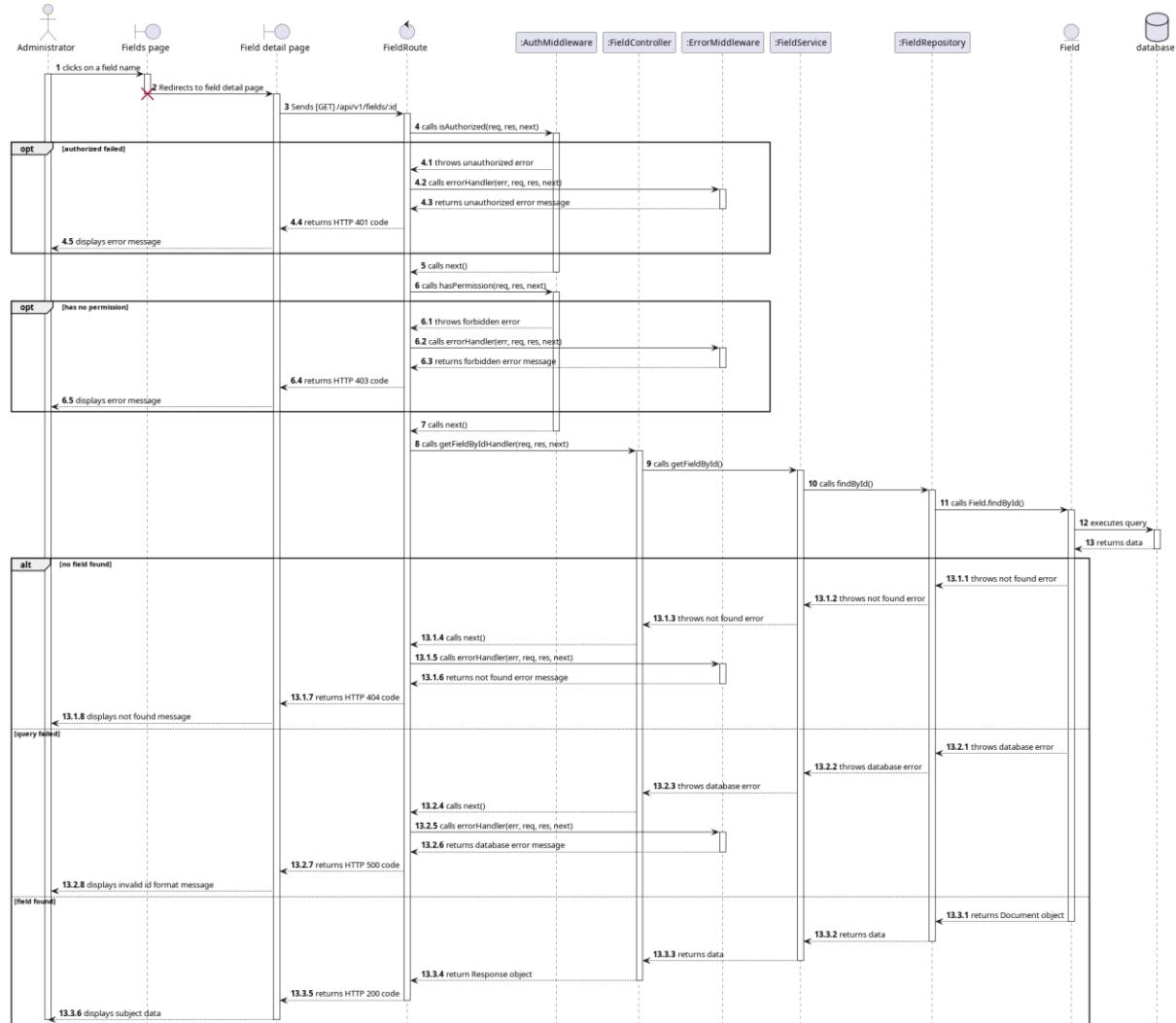


Figure 3.9.2.2: View field detail¹⁵⁵

¹⁵⁵ [view field detail.png](#)

3.9.3 Create field

3.9.3.1 Class diagram

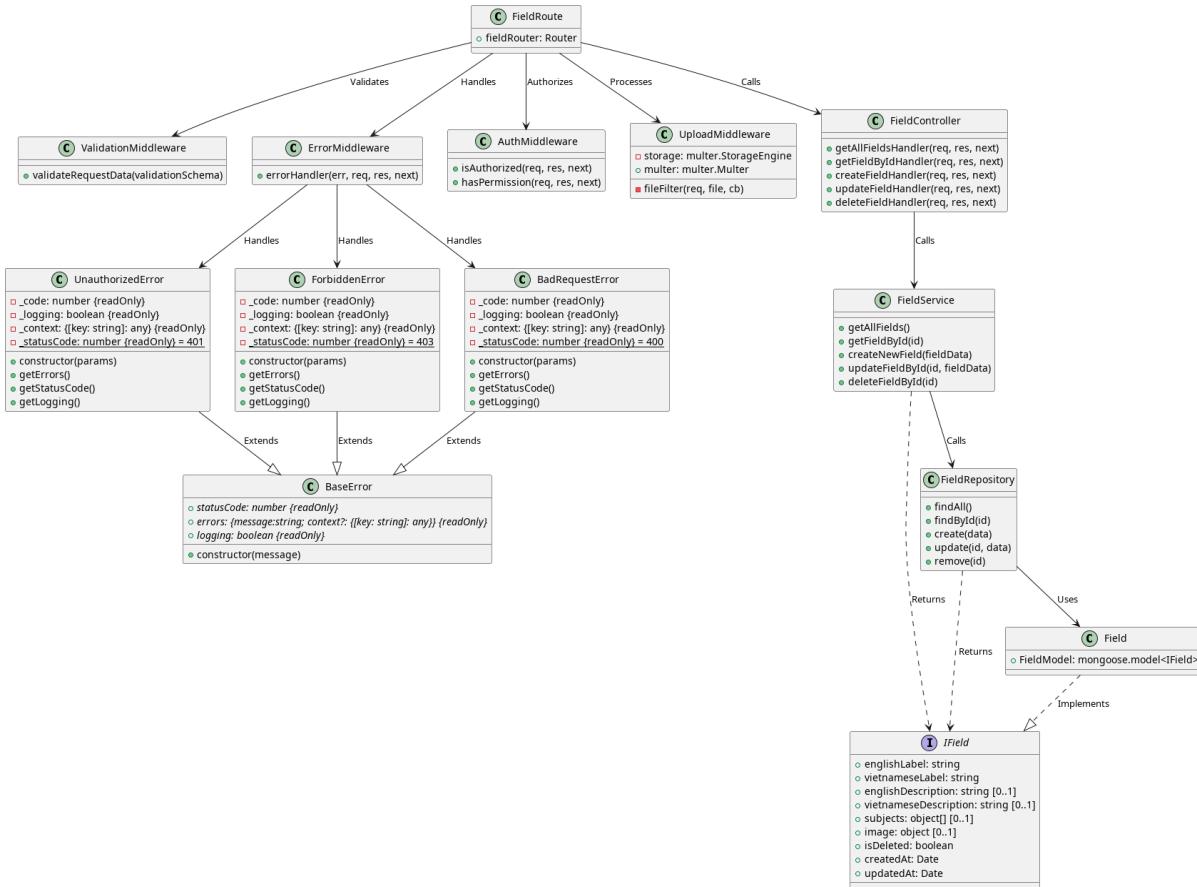


Figure 3.9.3.1: Create field¹⁵⁶

3.10.3.2 Sequence diagram

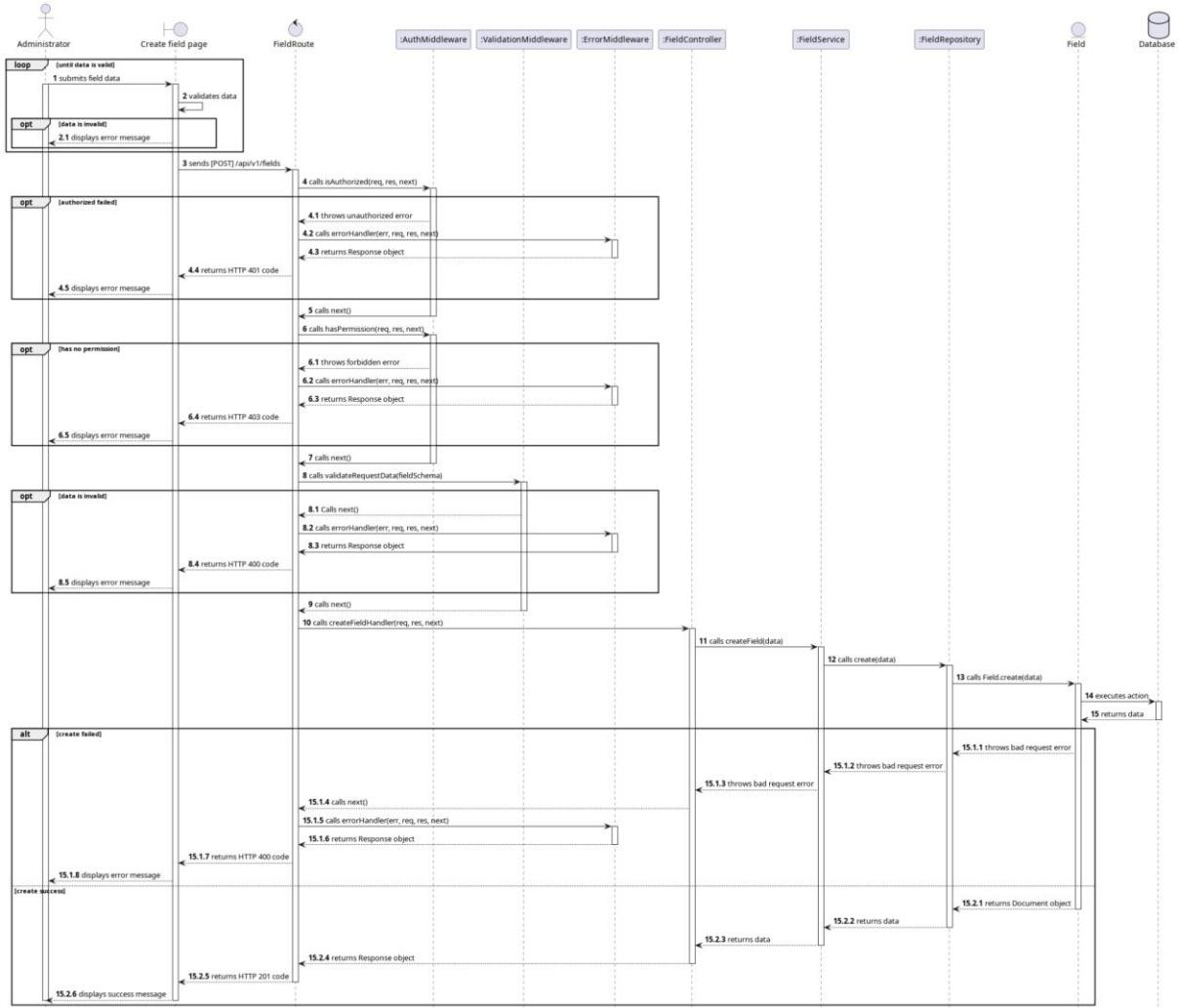


Figure 3.9.3.2: Create field¹⁵⁷

¹⁵⁷ [create fields.png](#)

3.9.4 Update field

3.9.4.1 Class diagram

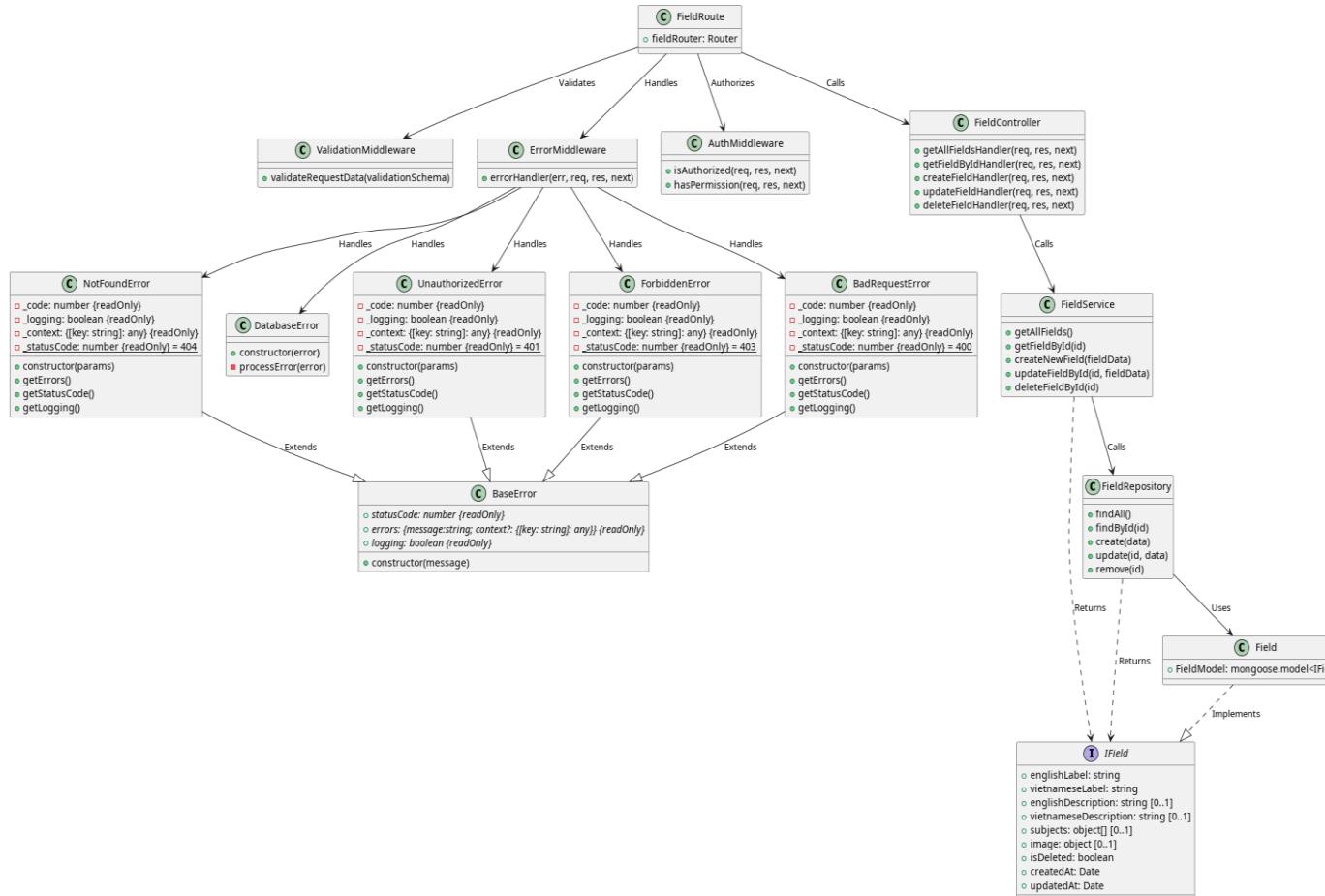


Figure 3.9.4.1: Update field¹⁵⁸

158 [update field.png](#)

3.9.4.2 Sequence diagram

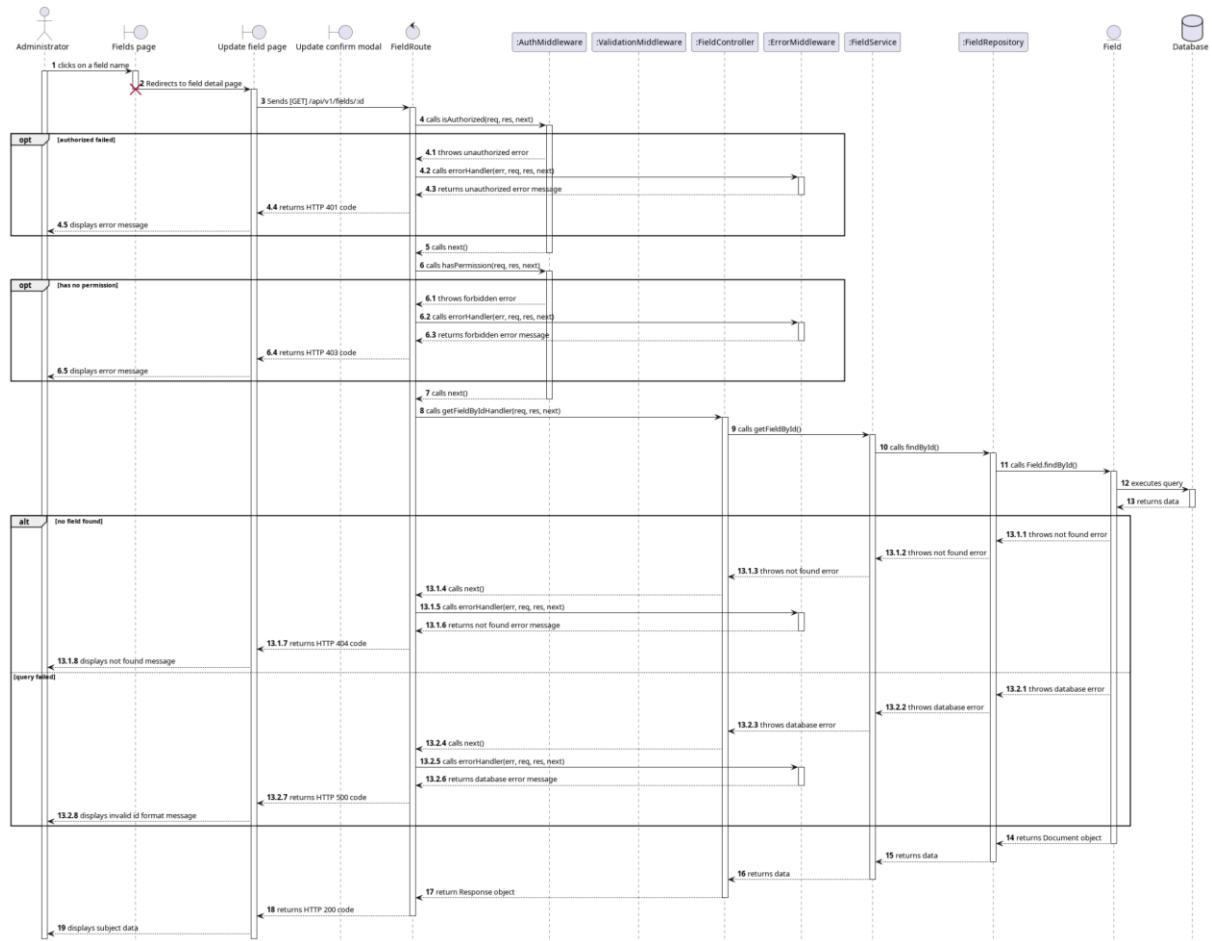


Figure 3.9.4.2.a: Get field detail¹⁵⁹

¹⁵⁹ [view field detail.png](#)

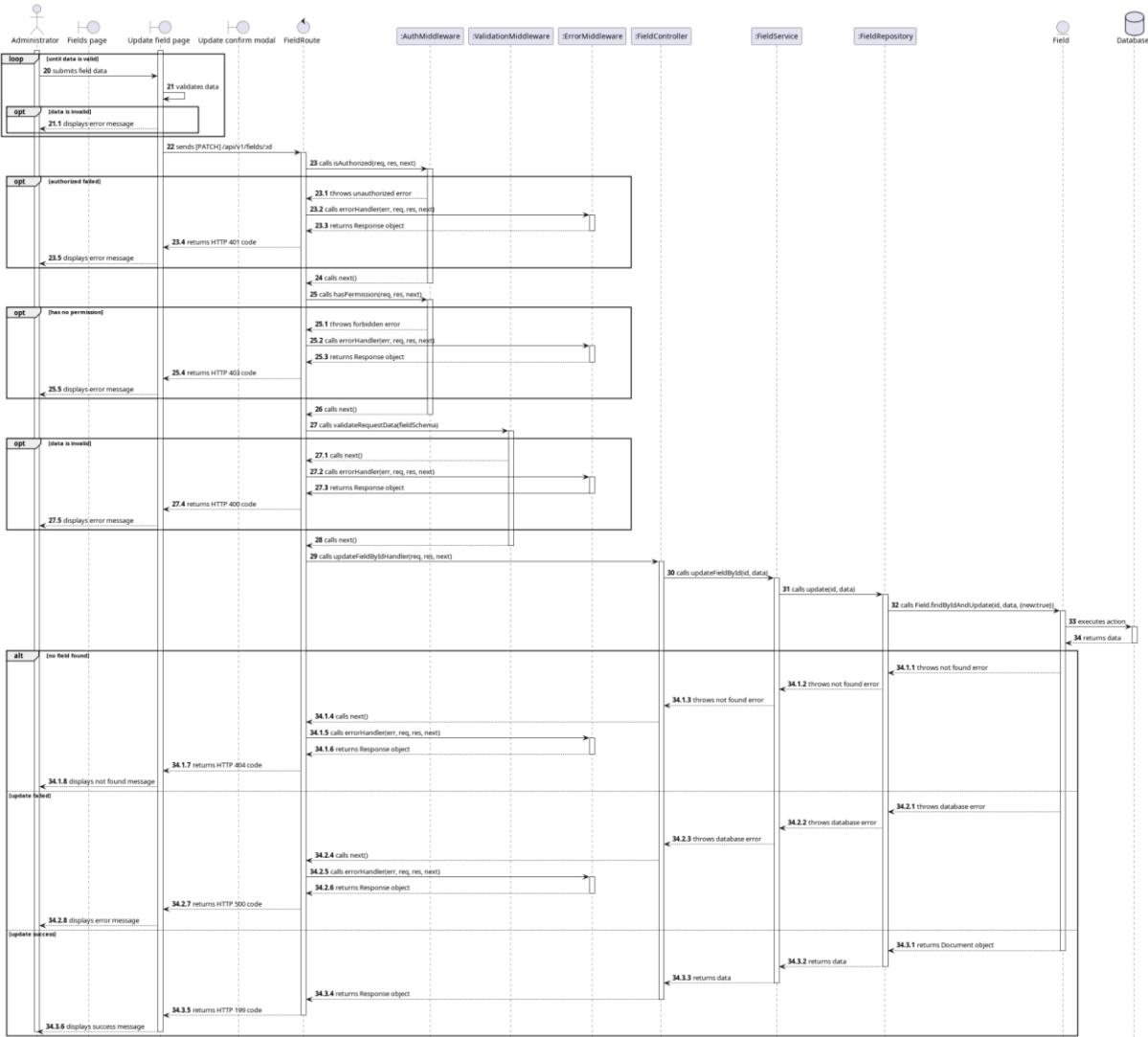


Figure 3.9.4.2.b: Update field¹⁶⁰

3.9.5 Delete field

3.9.5.1 Class diagram

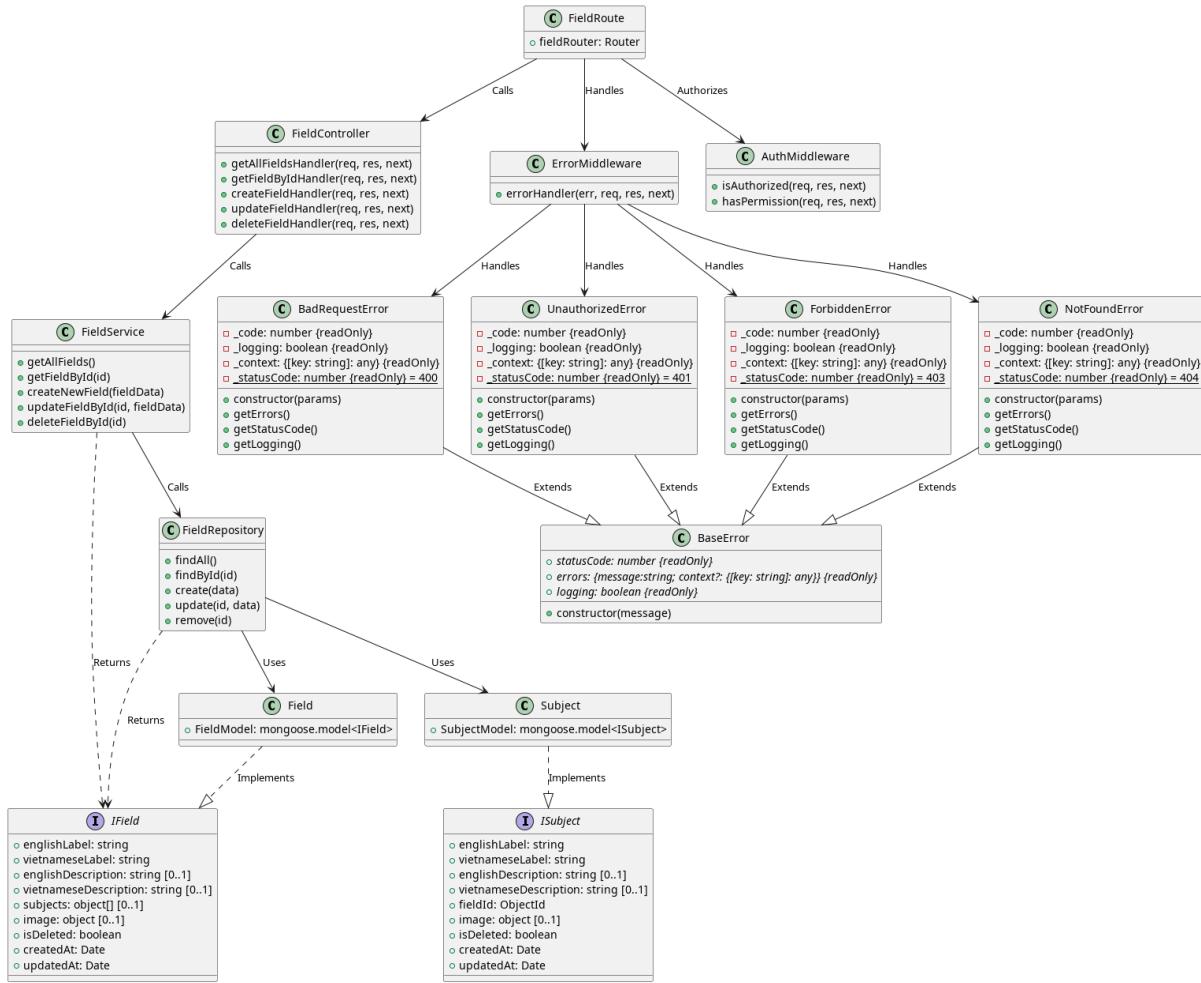


Figure 3.9.5.1: Delete field¹⁶¹

¹⁶¹ [delete field.png](#)

3.9.5.2 Sequence diagram

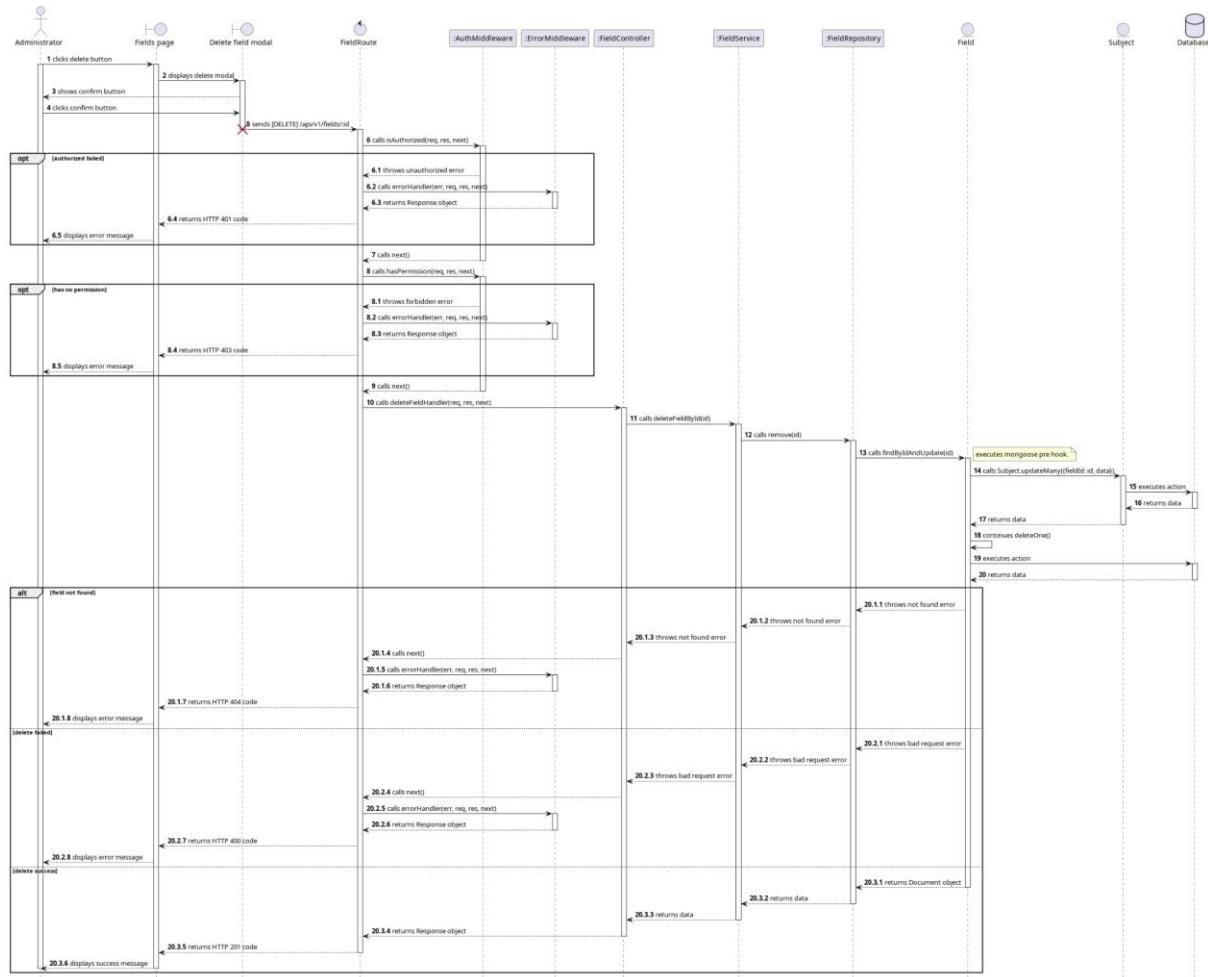


Figure 3.9.5.2: Delete field¹⁶²

¹⁶² [delete fields.png](#)

3.10 Subject Management

3.10.1 View subjects

3.10.1.1 Class diagram

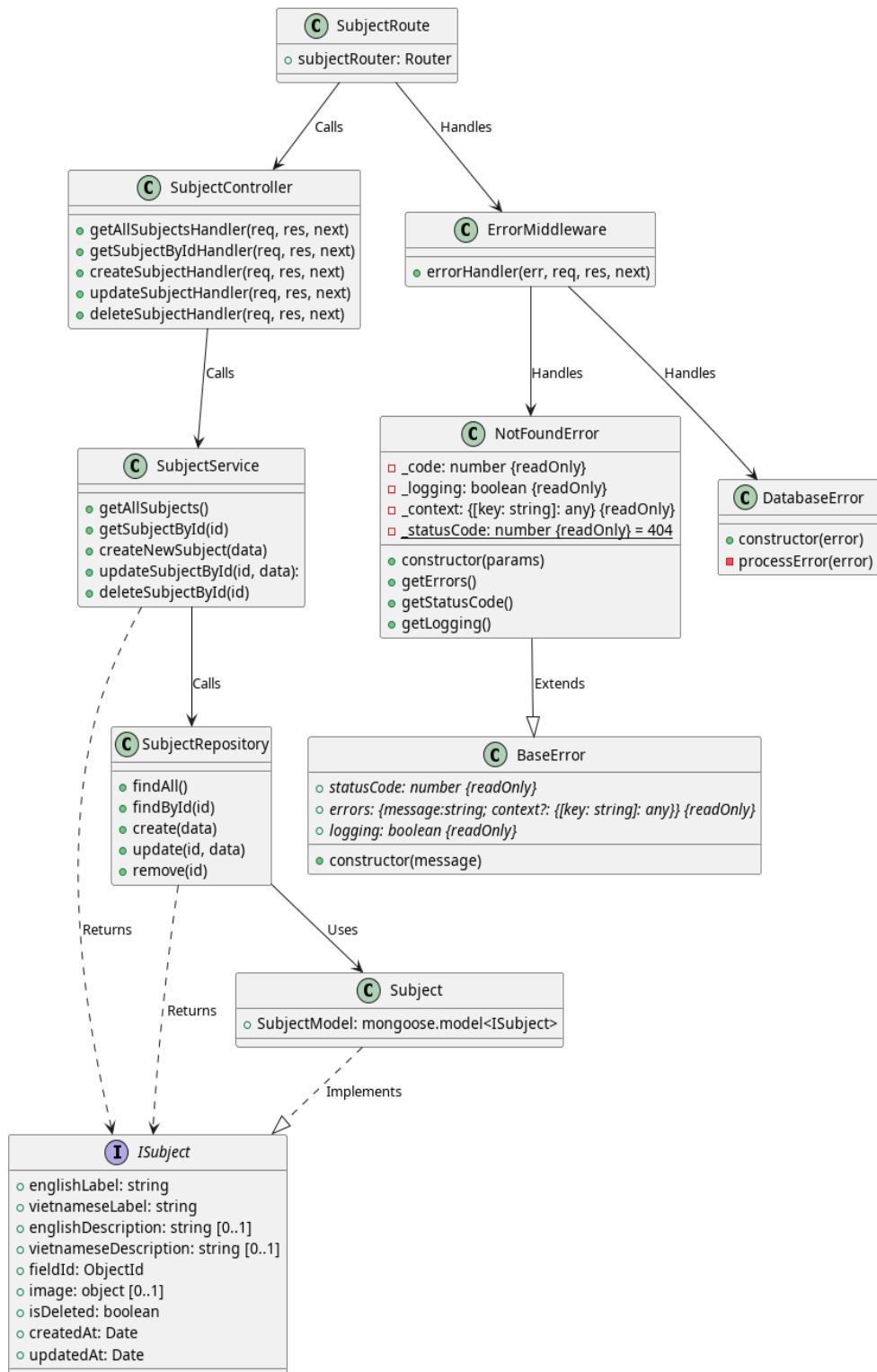


Figure 3.10.1.1: View subjects¹⁶³

¹⁶³ [view all subjects.png](#)

3.10.1.2 Sequence diagram

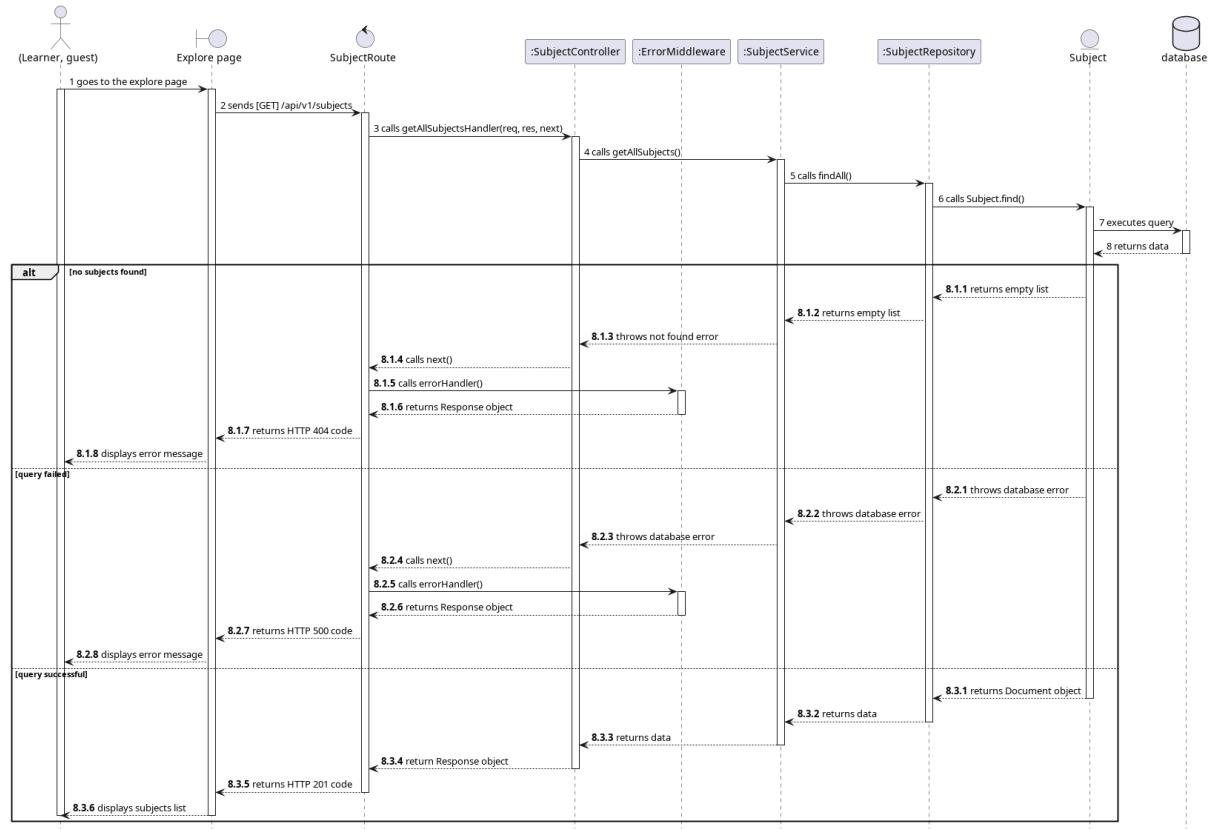


Figure 3.10.1.2.a: Learners and Guests view subjects¹⁶⁴

¹⁶⁴ [view subjects.png](#)

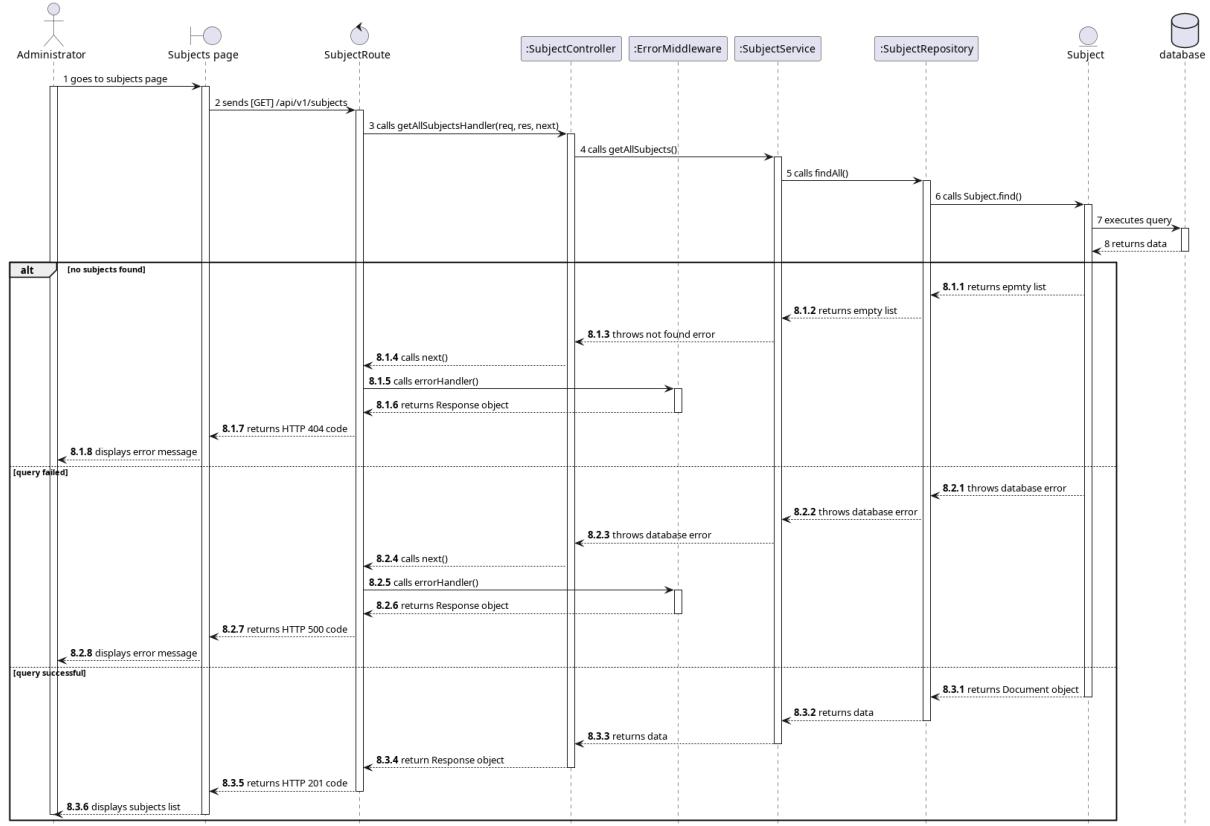


Figure 3.10.1.2.b: Administrator view subjects¹⁶⁵

165 [view subjects.png](#)

3.10.2 View subject details

3.10.2.1 Class diagram

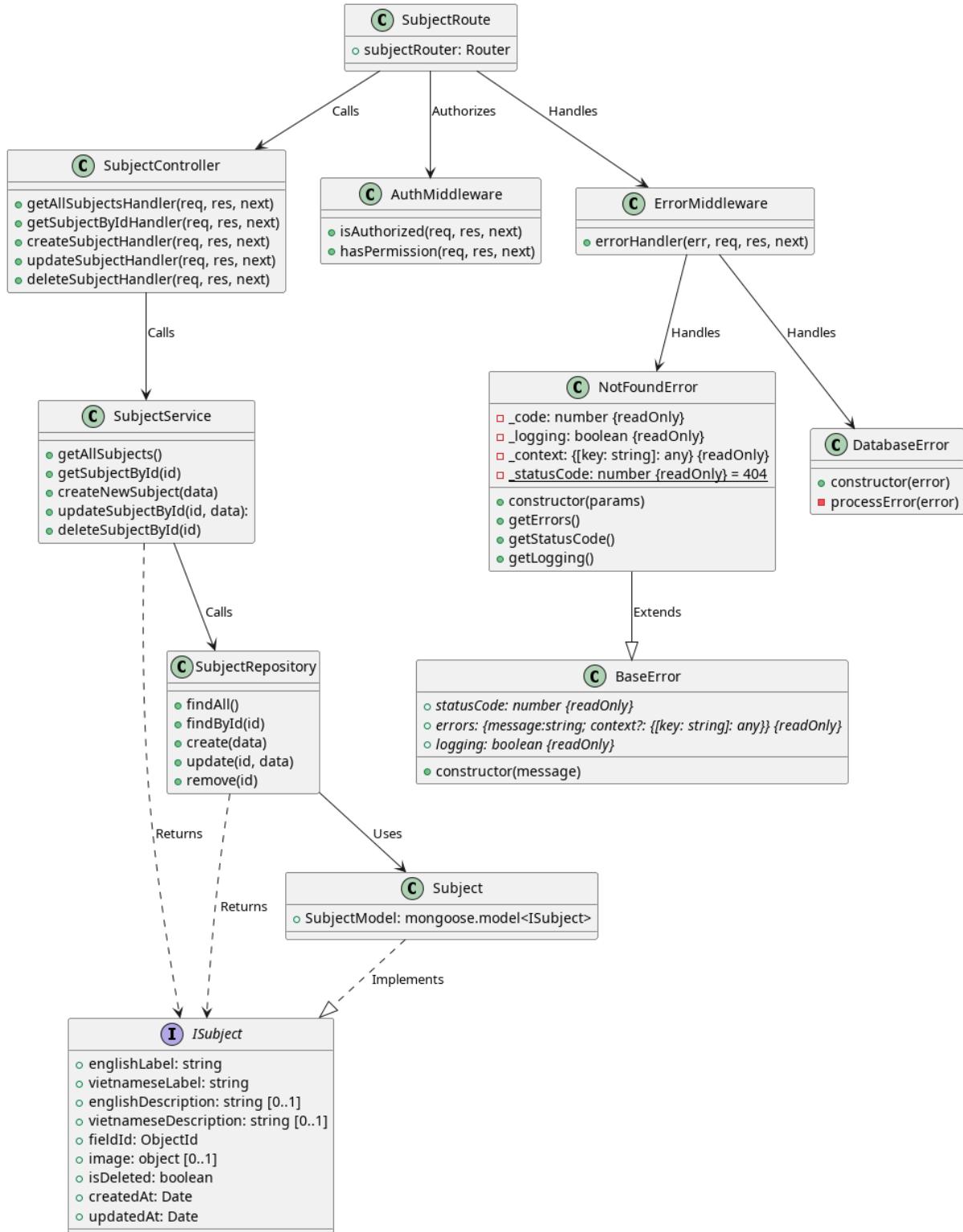


Figure 3.10.2.1: View subject detail¹⁶⁶

¹⁶⁶ [view subject detail.png](#)

3.10.2.2 Sequence diagram

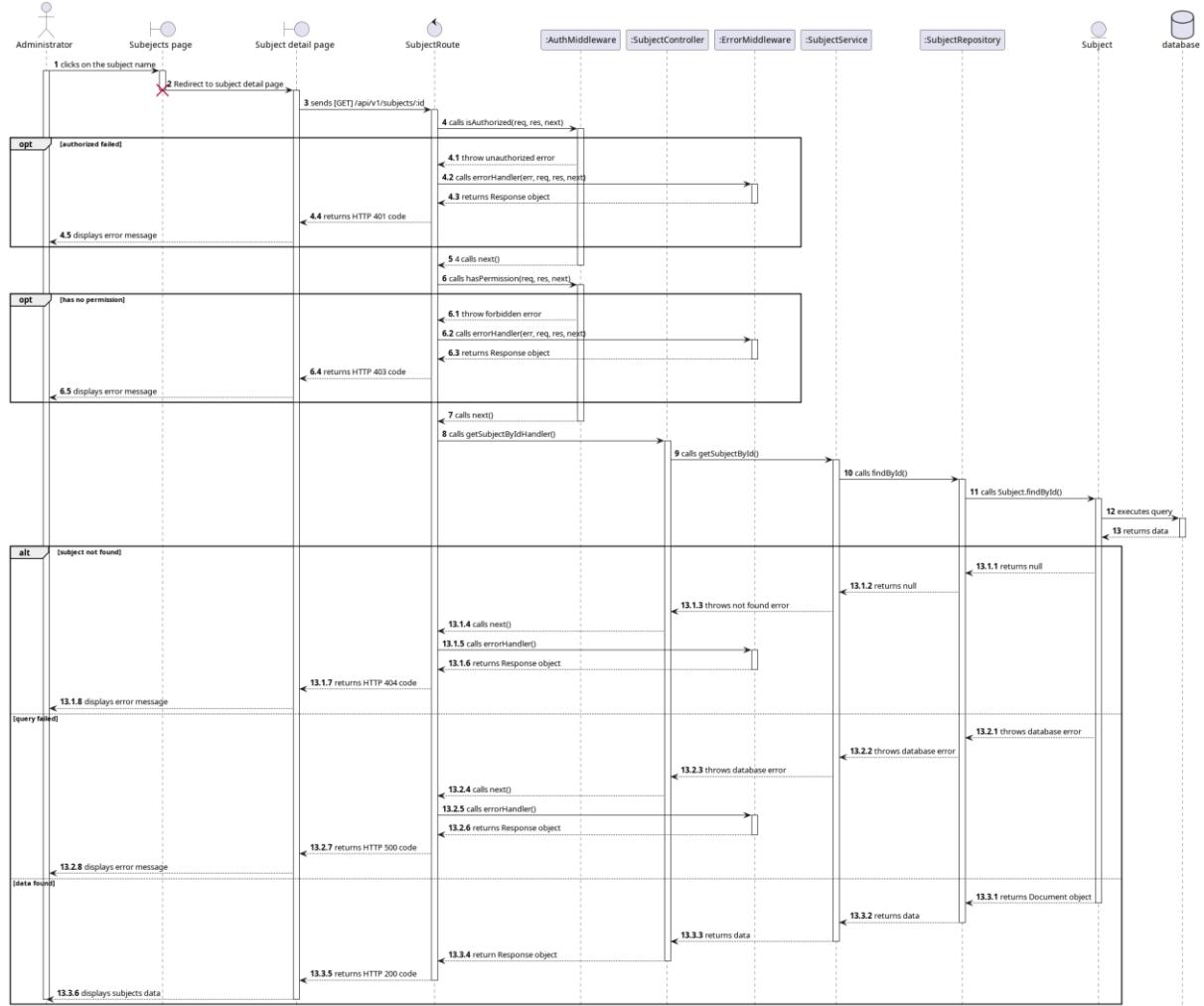


Figure 3.10.2.2: View subject detail¹⁶⁷

¹⁶⁷ [view subject detail.png](#)

3.10.3 Create subject

3.10.3.1 Class diagram

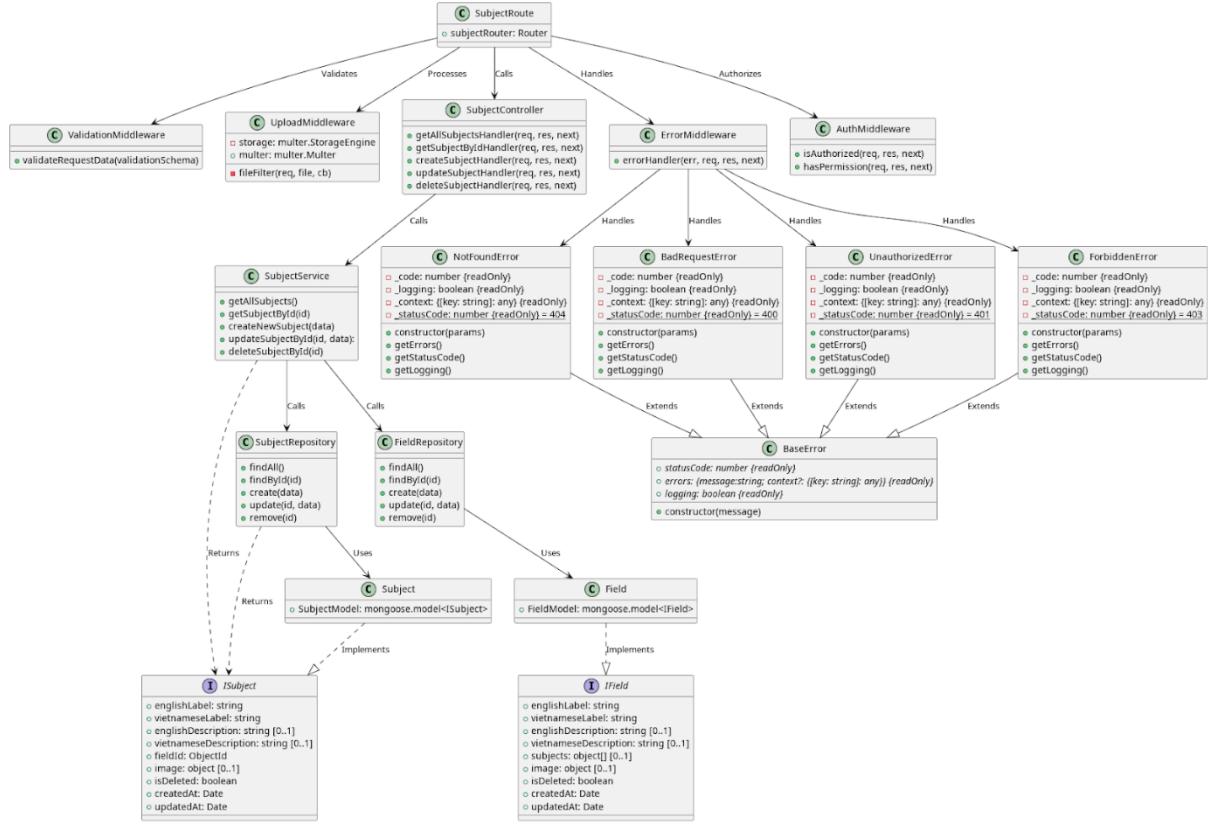


Figure 3.10.3.1: Create subject¹⁶⁸

¹⁶⁸ [create subject.png](#)

3.10.3.2 Sequence diagram

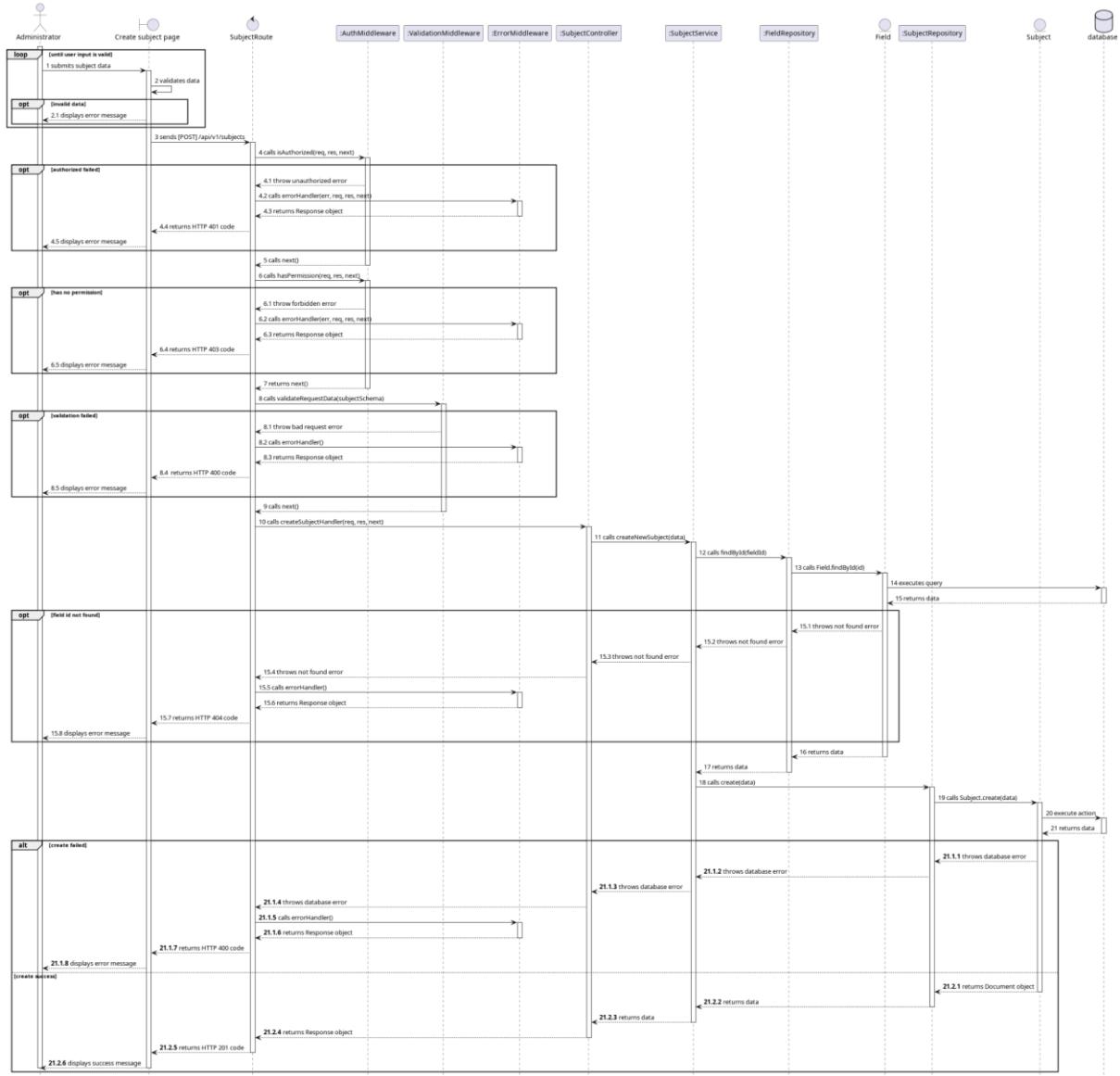


Figure 3.10.3.2: Create subject¹⁶⁹

¹⁶⁹ [create subject.png](#)

3.10.4 Update subject

3.10.4.1 Class diagram

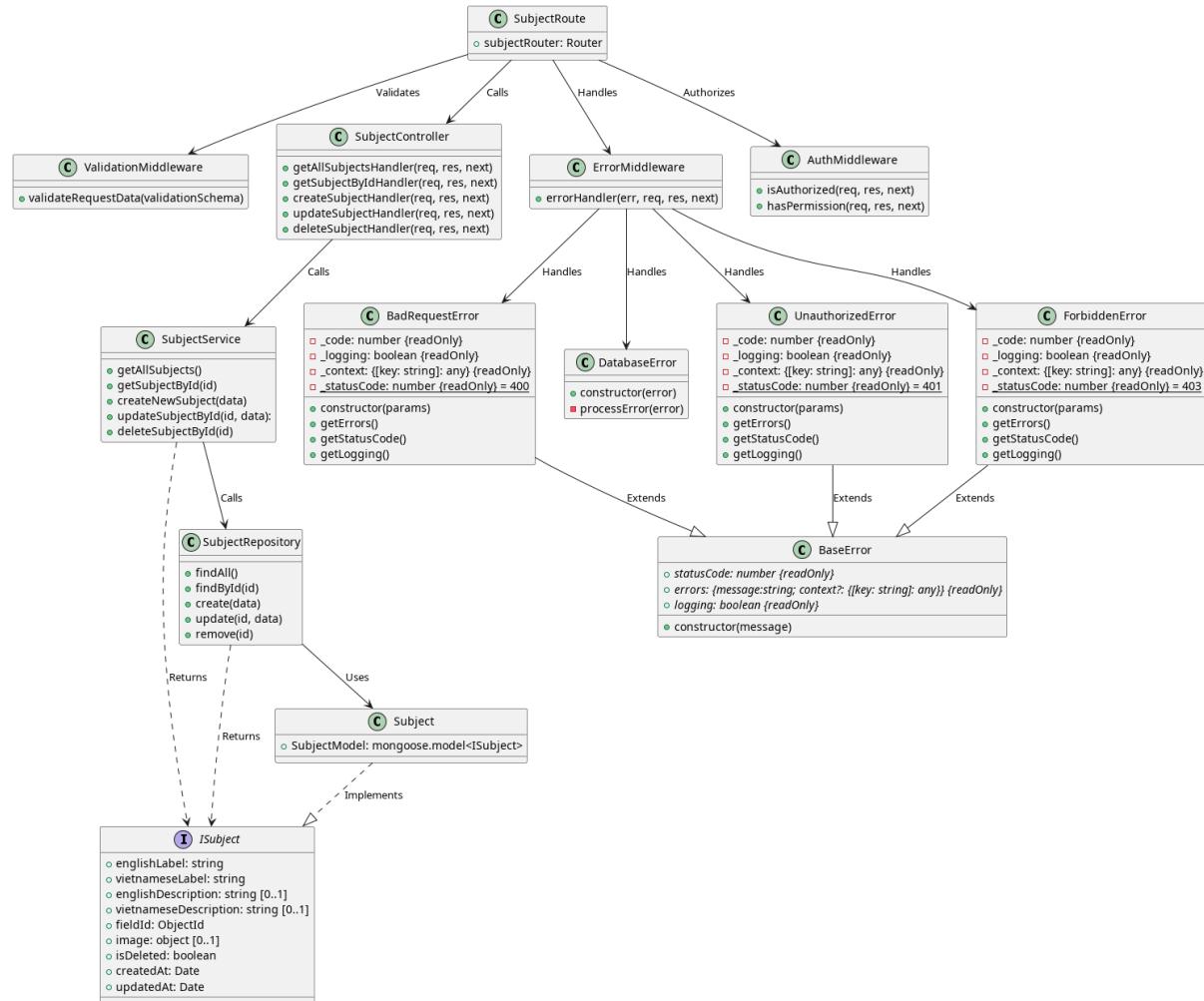


Figure 3.10.4.2: Update subject¹⁷⁰

¹⁷⁰ [update subject.png](#)

3.10.4.2 Sequence diagram

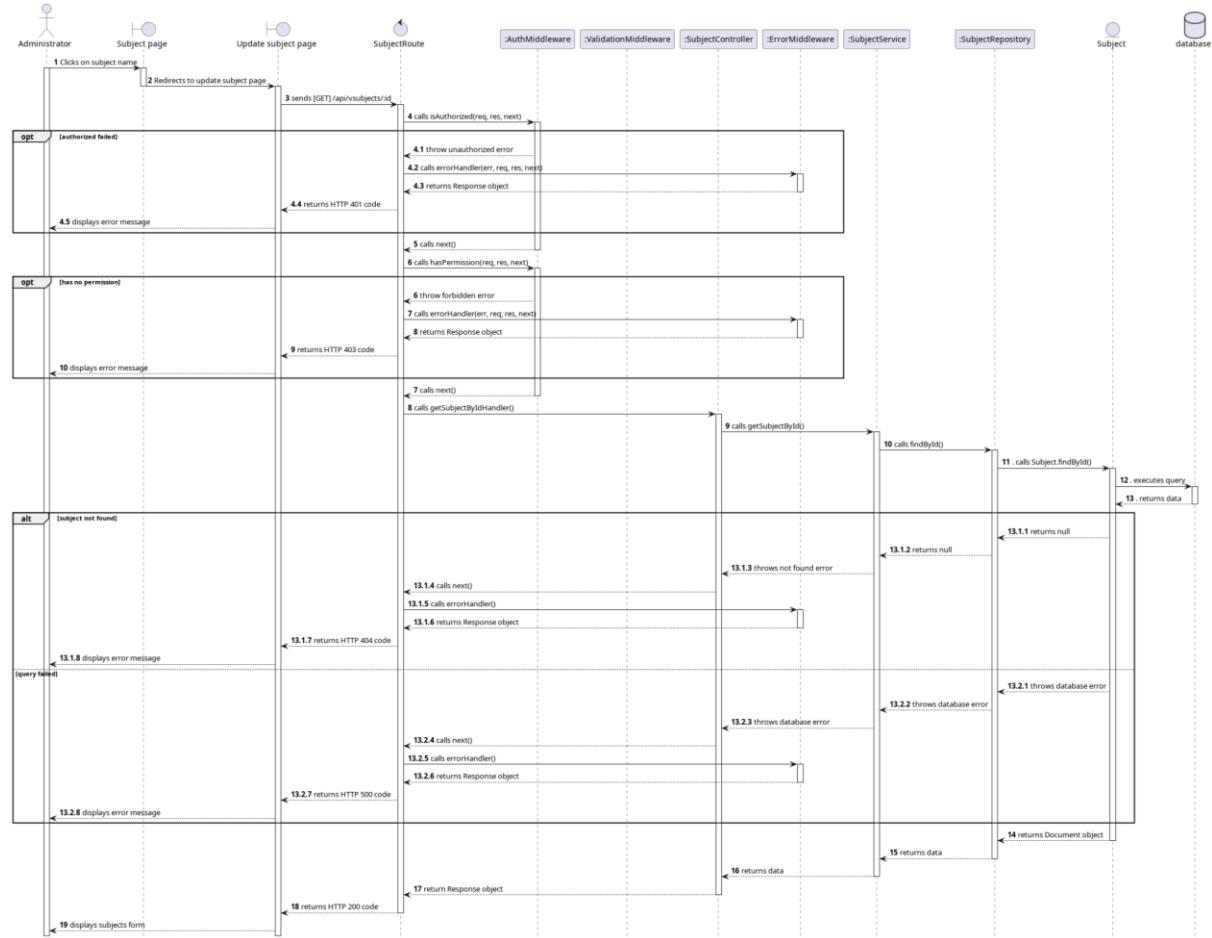


Figure 3.10.4.2.a: Get subject detail¹⁷¹

¹⁷¹ [view subject detail.png](#)

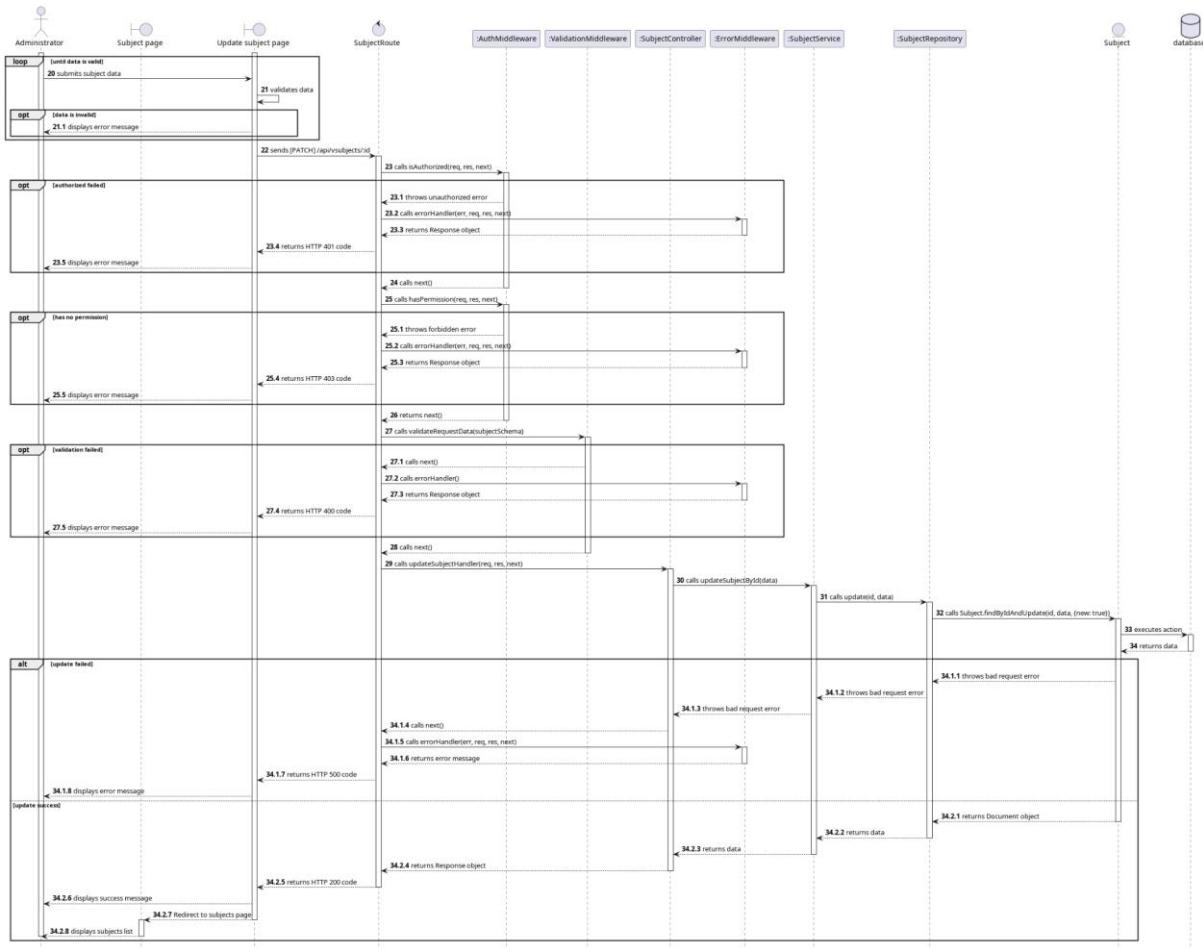


Figure 3.10.4.2.b: Update subject¹⁷²

¹⁷² update subject.png

3.10.5 Delete subject

3.10.5.1 Class diagram

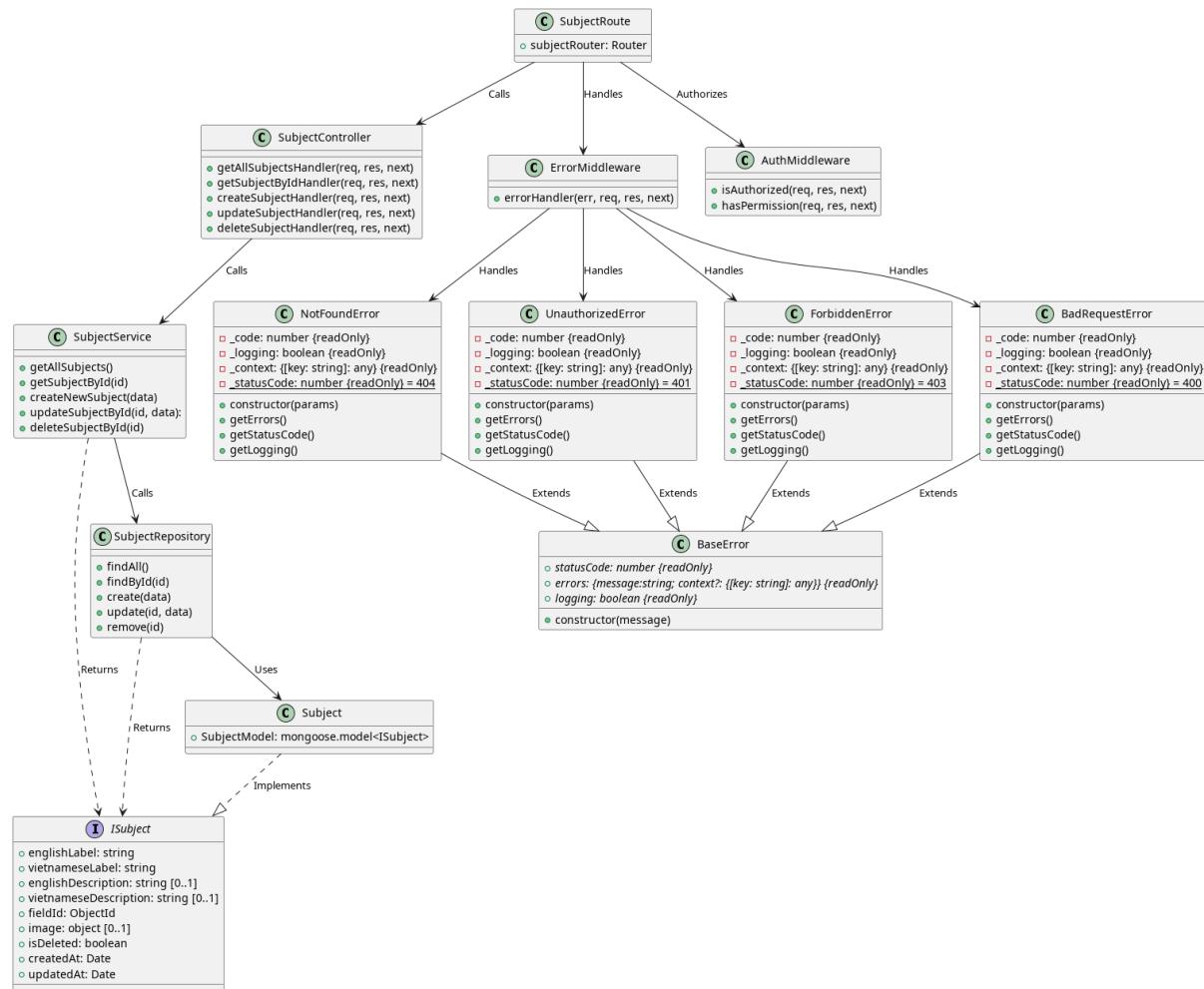


Figure 3.10.5.1: Delete subject¹⁷³

¹⁷³ [delete subject.png](#)

3.10.5.2 Sequence diagram

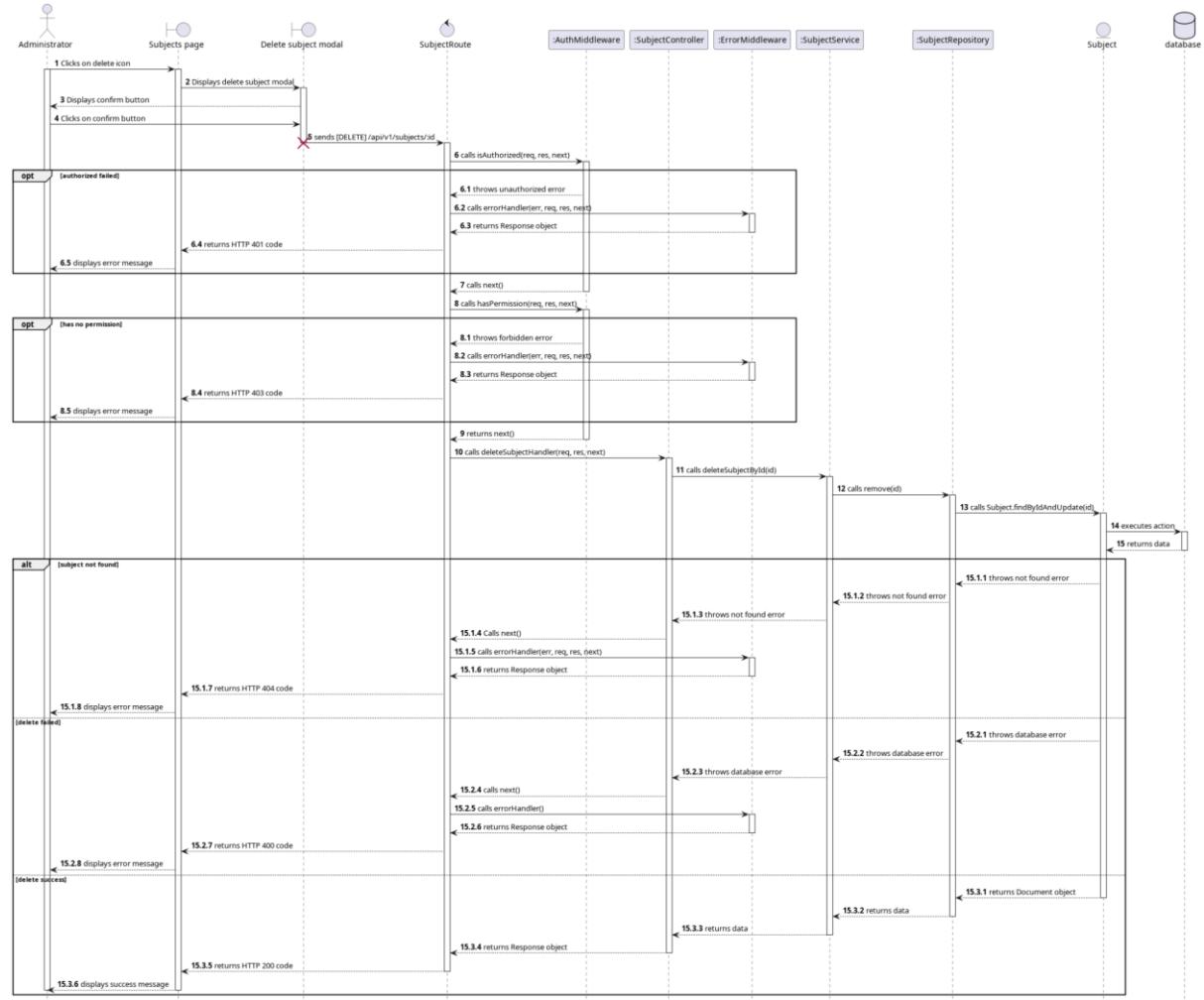


Figure 3.10.5.2: Delete subject¹⁷⁴

¹⁷⁴ [delete subject.png](#)

3.11 Following Management

3.11.1 Follow account

3.11.1.1 Class diagram

3.11.1.1.1 Follow account

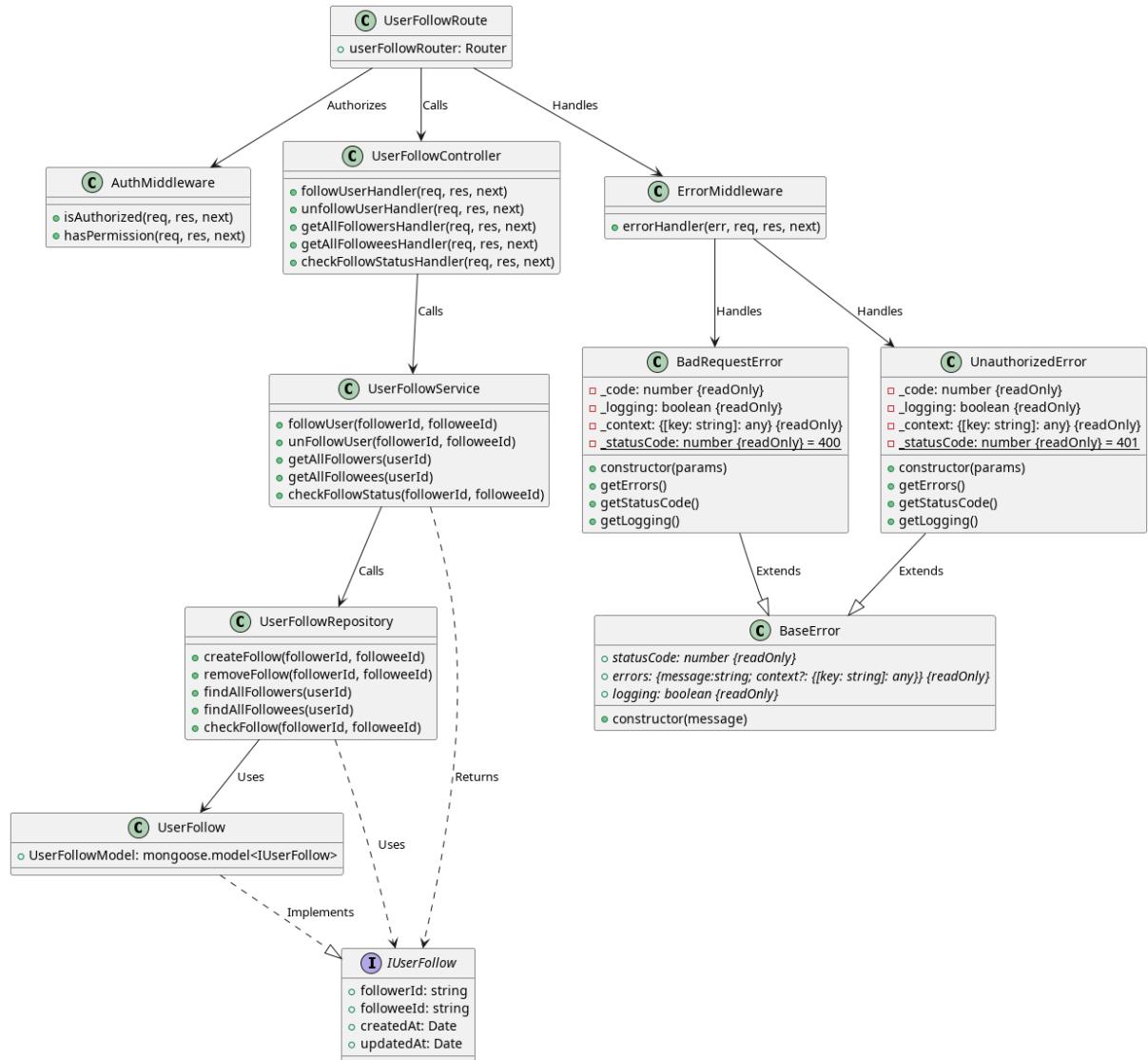


Figure 3.11.1.1.1: Follow account¹⁷⁵

¹⁷⁵ [follow.png](#)

3.11.1.1.2 Unfollow account

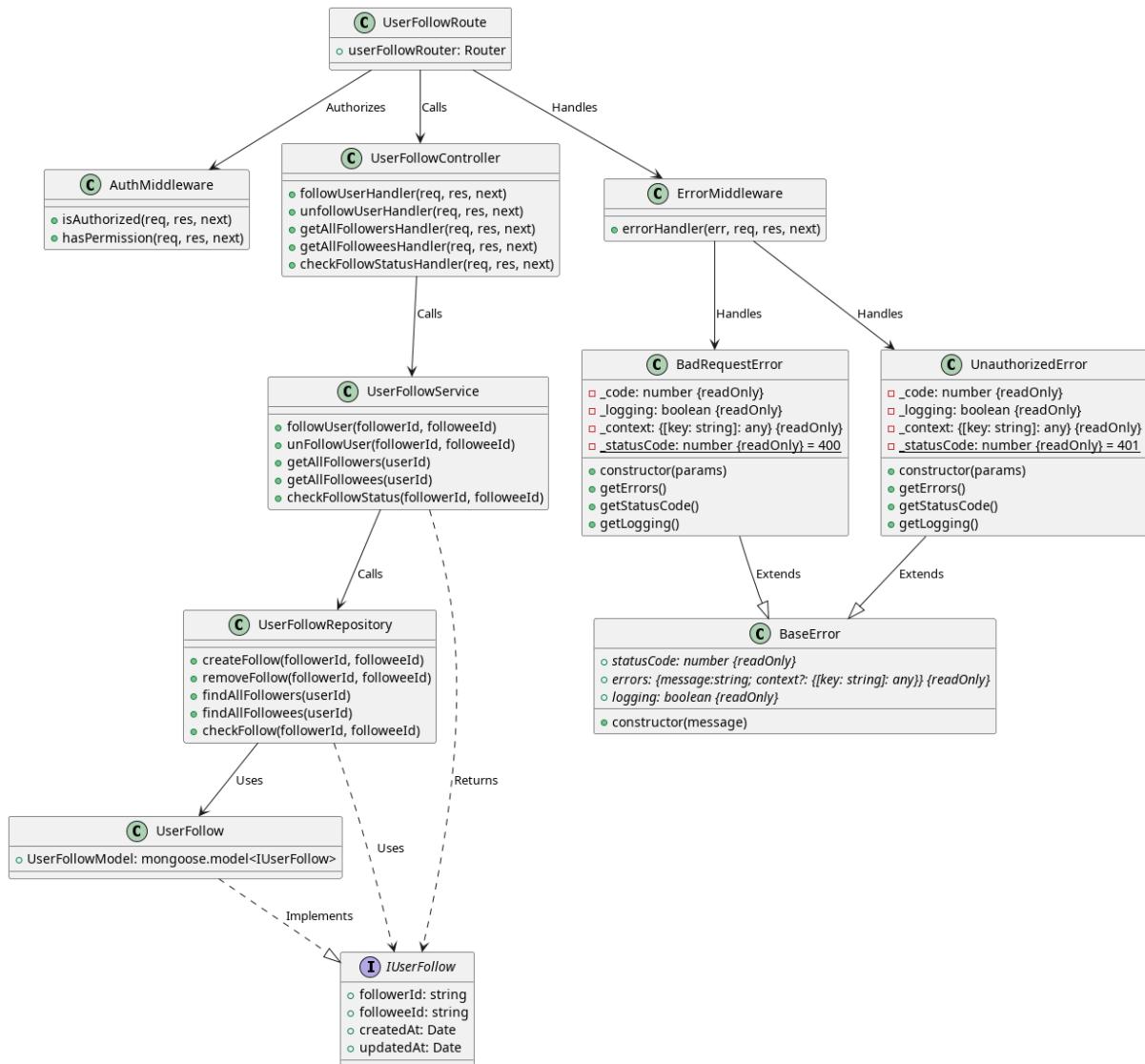


Figure 3.11.1.1.2: Unfollow account¹⁷⁶

¹⁷⁶ [unfollow.png](#)

3.11.1.2 Sequence diagram

3.11.1.2.1 Follow account

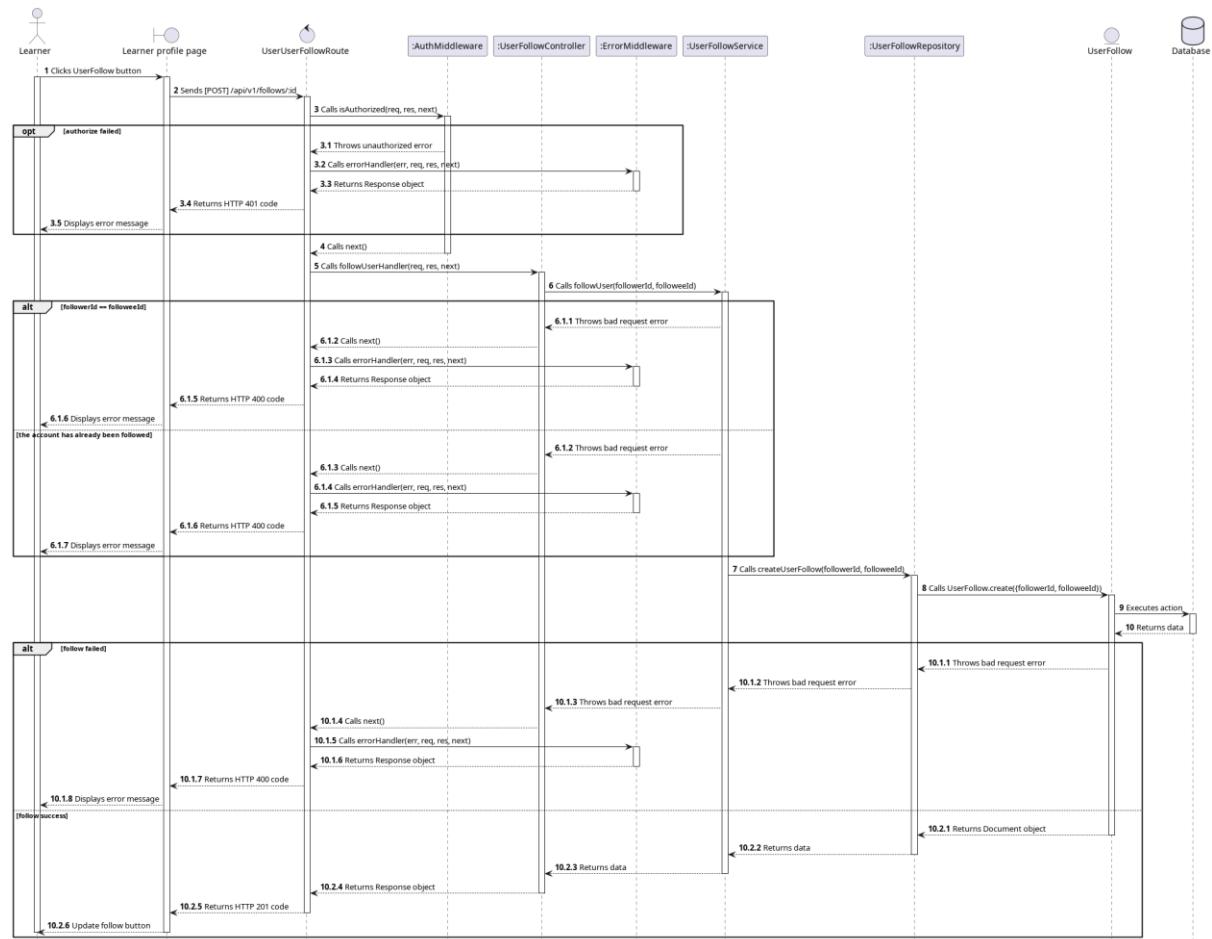


Figure 3.11.1.2.1: Follow account¹⁷⁷

¹⁷⁷ [follow.png](#)

3.11.1.2.2 Unfollow account

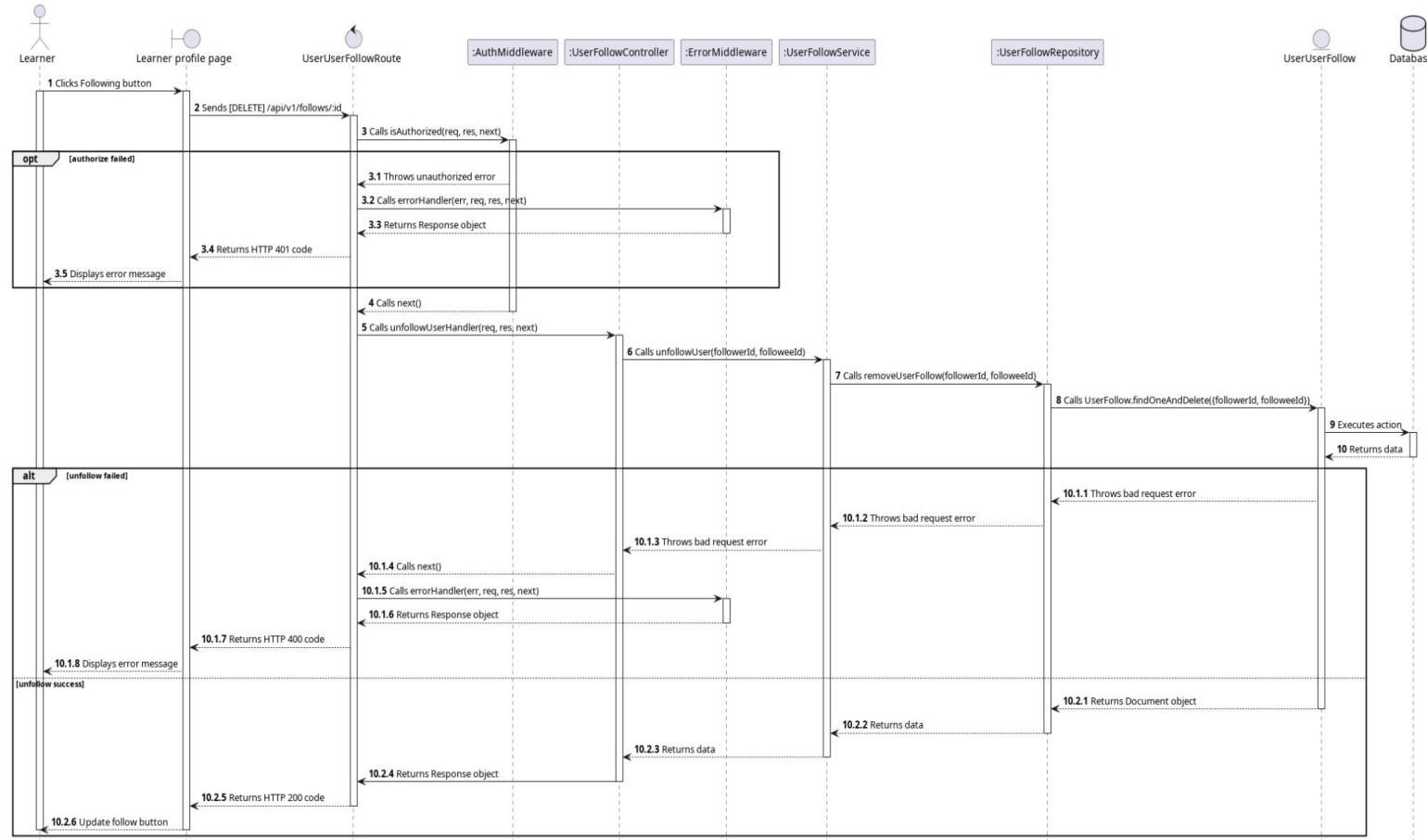


Figure 3.11.1.2.2: Unfollow account¹⁷⁸

178 [unfollow.png](#)

3.11.2 View follow detail

3.11.2.1 Class diagram

3.11.2.1.1 View followers

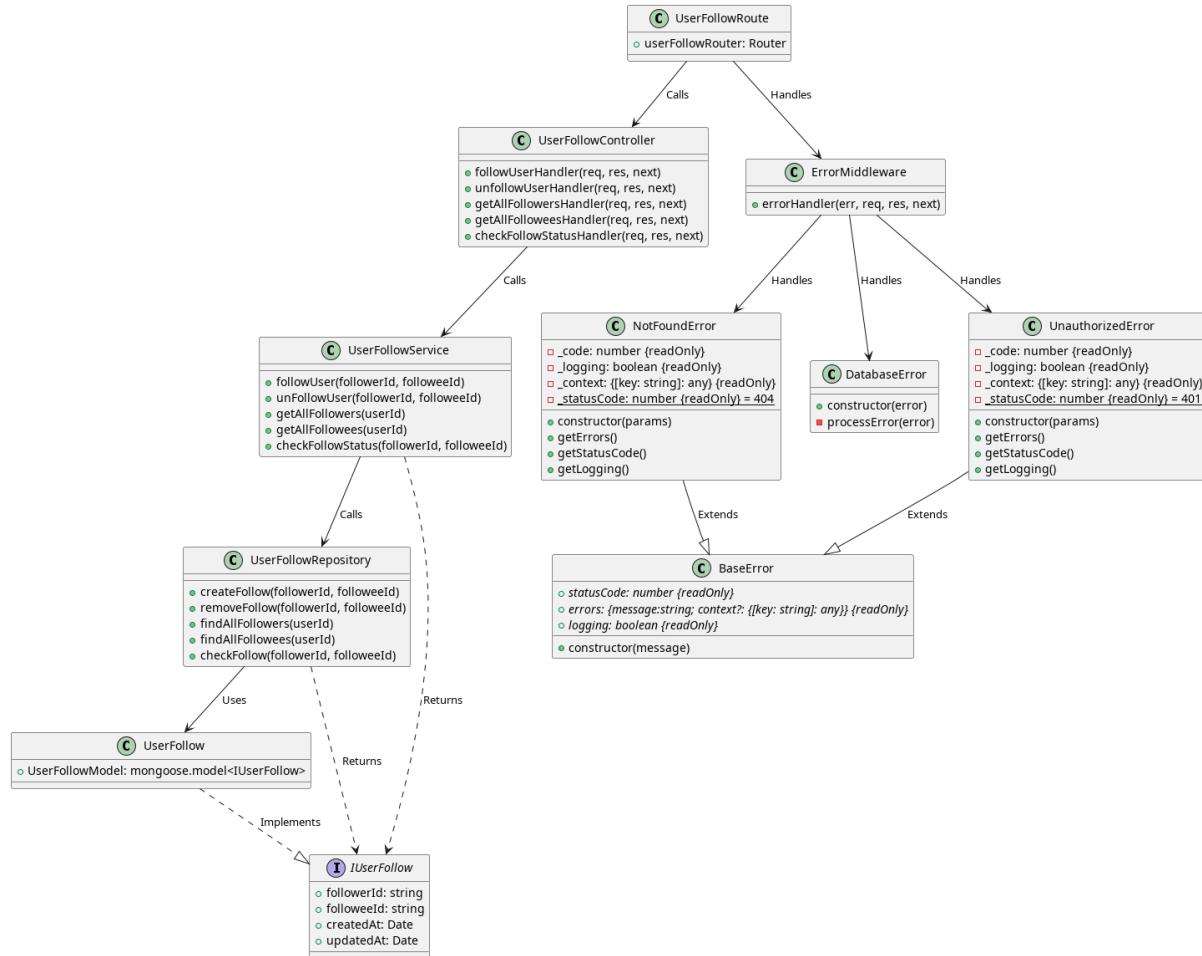


Figure 3.11.2.1.1: View followers¹⁷⁹

¹⁷⁹ [view followers.png](#)

3.11.2.1.2 View followings

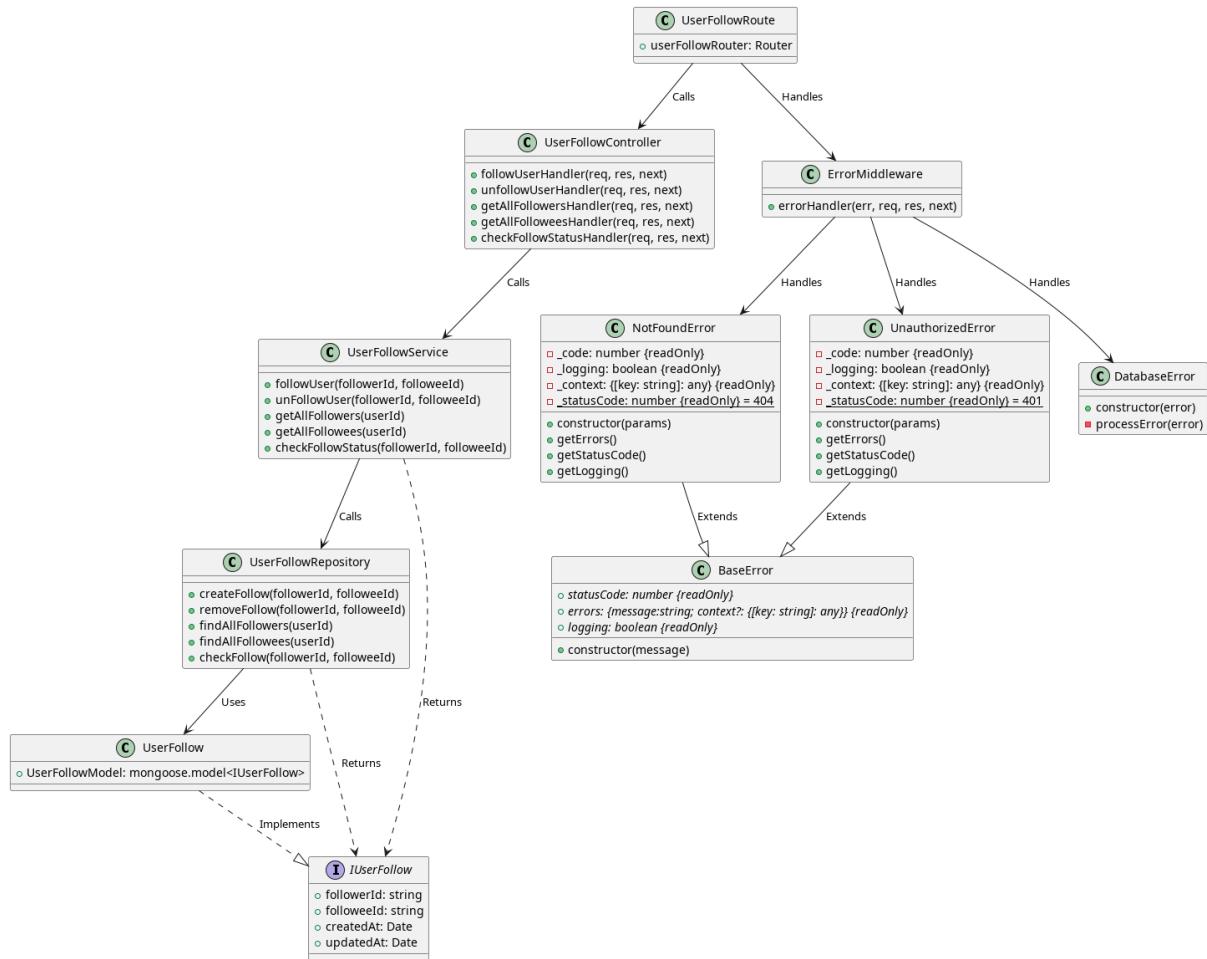


Figure 3.11.2.1.2: View followings¹⁸⁰

¹⁸⁰ [view followings.png](#)

3.11.2.2 Sequence diagram

3.11.2.2.1 View followers

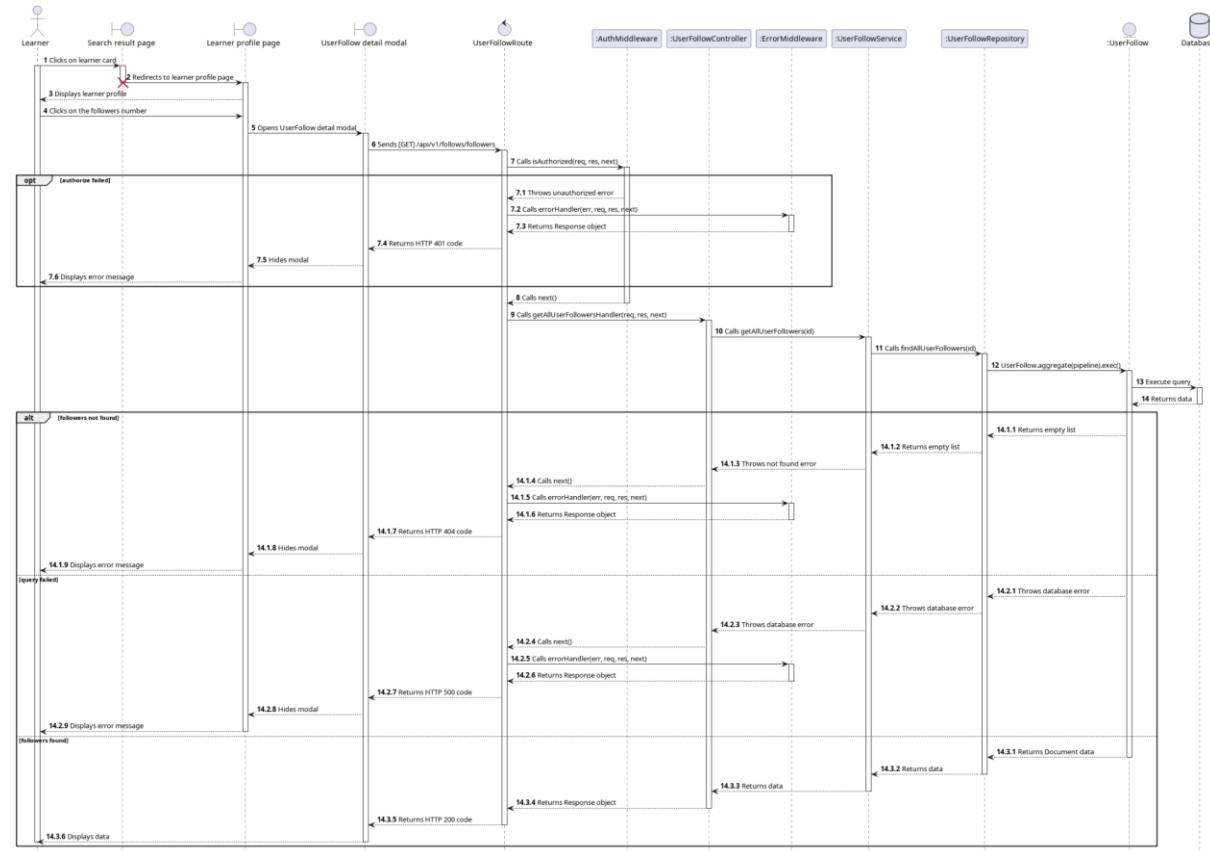


Figure 3.11.2.2.1.a: Learner view followers¹⁸¹

¹⁸¹ [view followers.png](#)

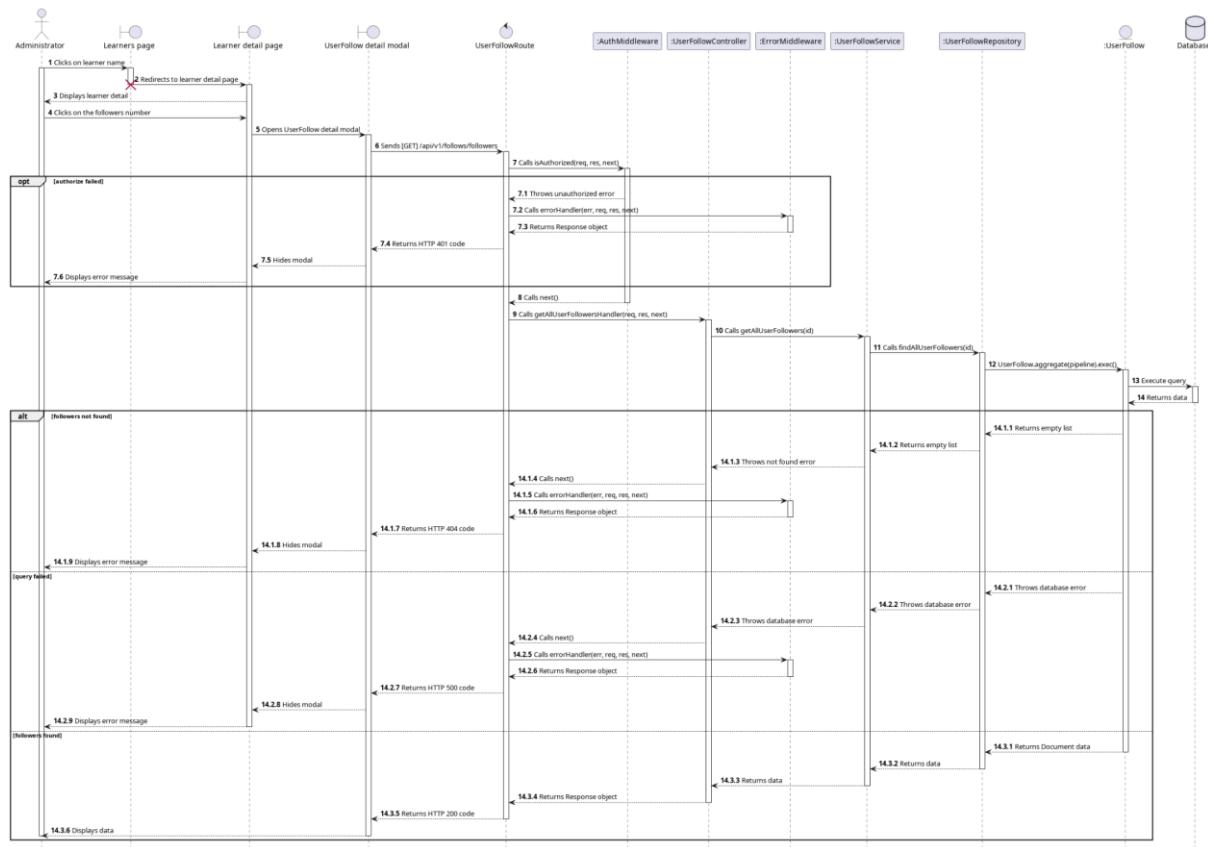


Figure 3.11.2.2.1.b: Administrator view followers¹⁸²

182 [view followers.png](#)

3.11.2.2 View followings

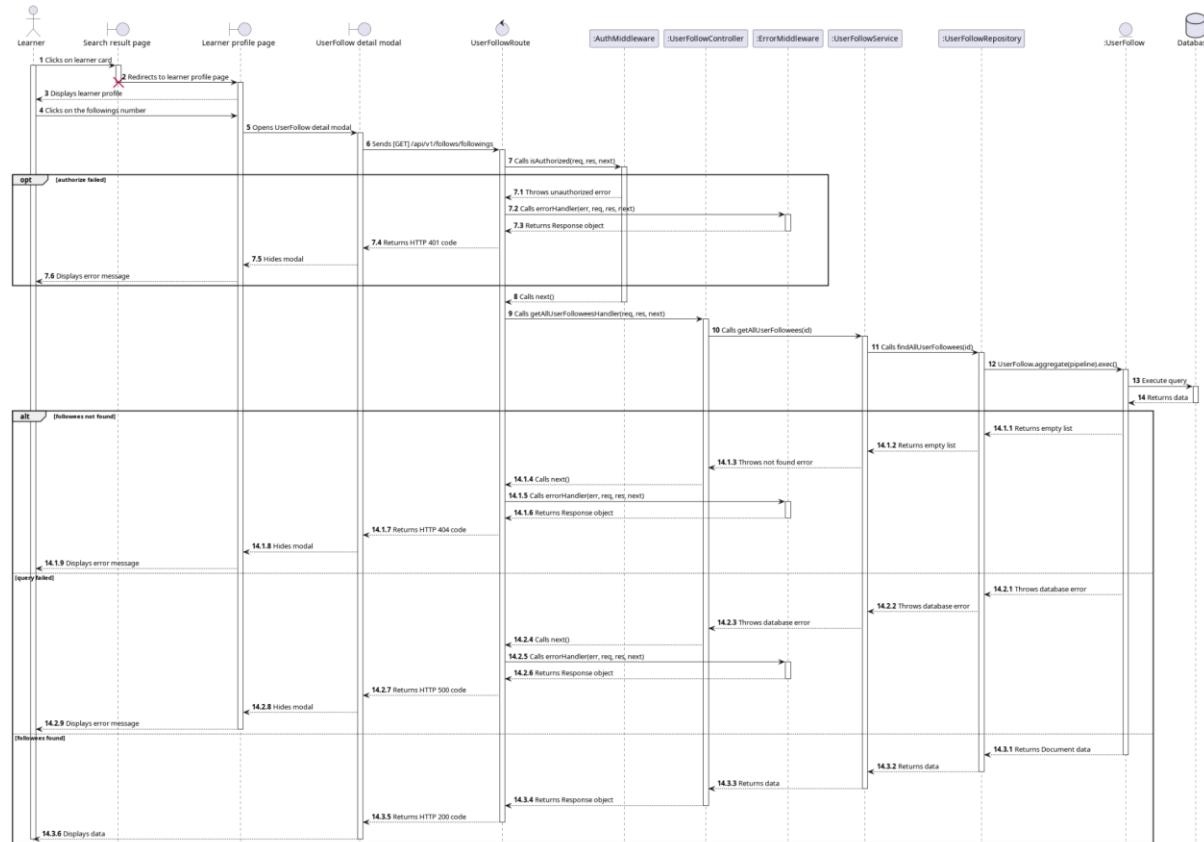


Figure 3.11.2.2.2.a: Learner view followings¹⁸³

¹⁸³ view followings.png

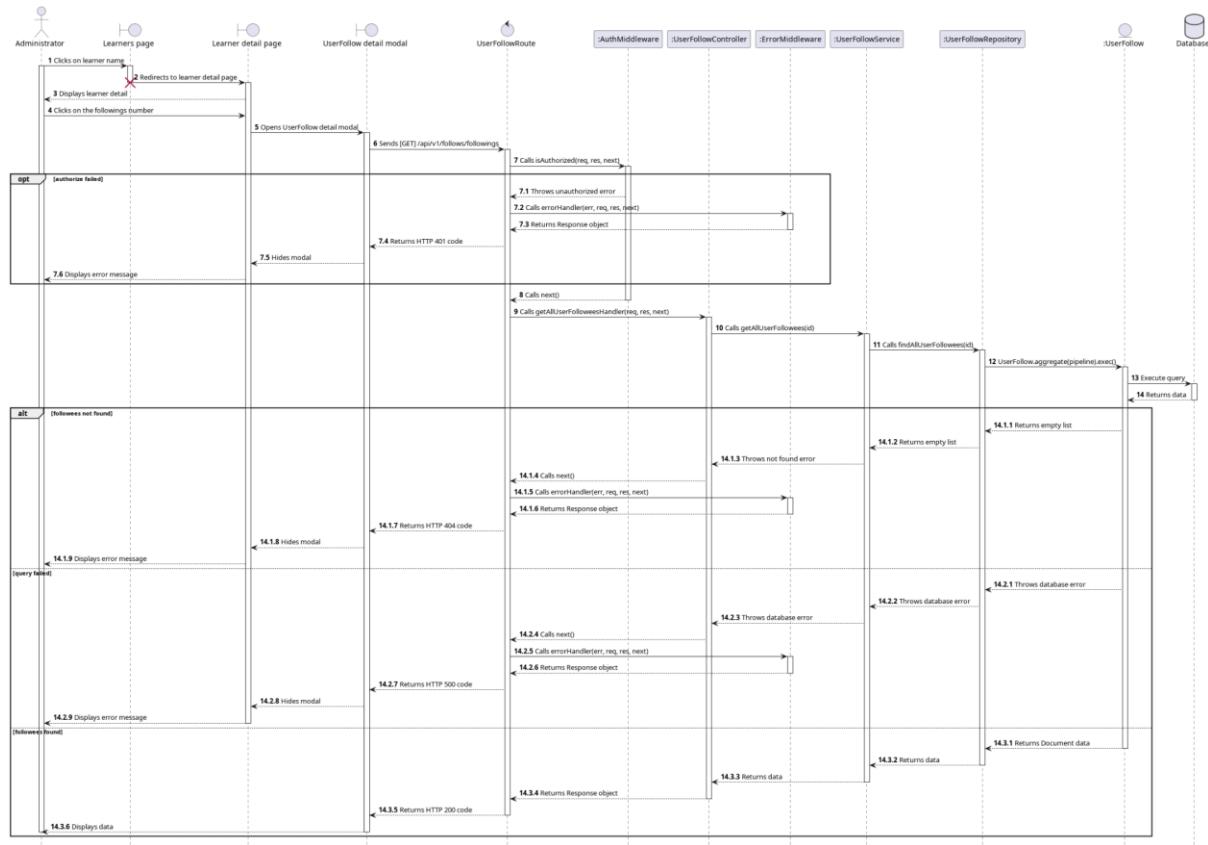


Figure 3.11.2.2.2.b: Administrator view followings¹⁸⁴

184 [view followings.png](#)

3.12 Question Set Progress Management

3.12.1 View progress

3.12.1.1 Class diagram

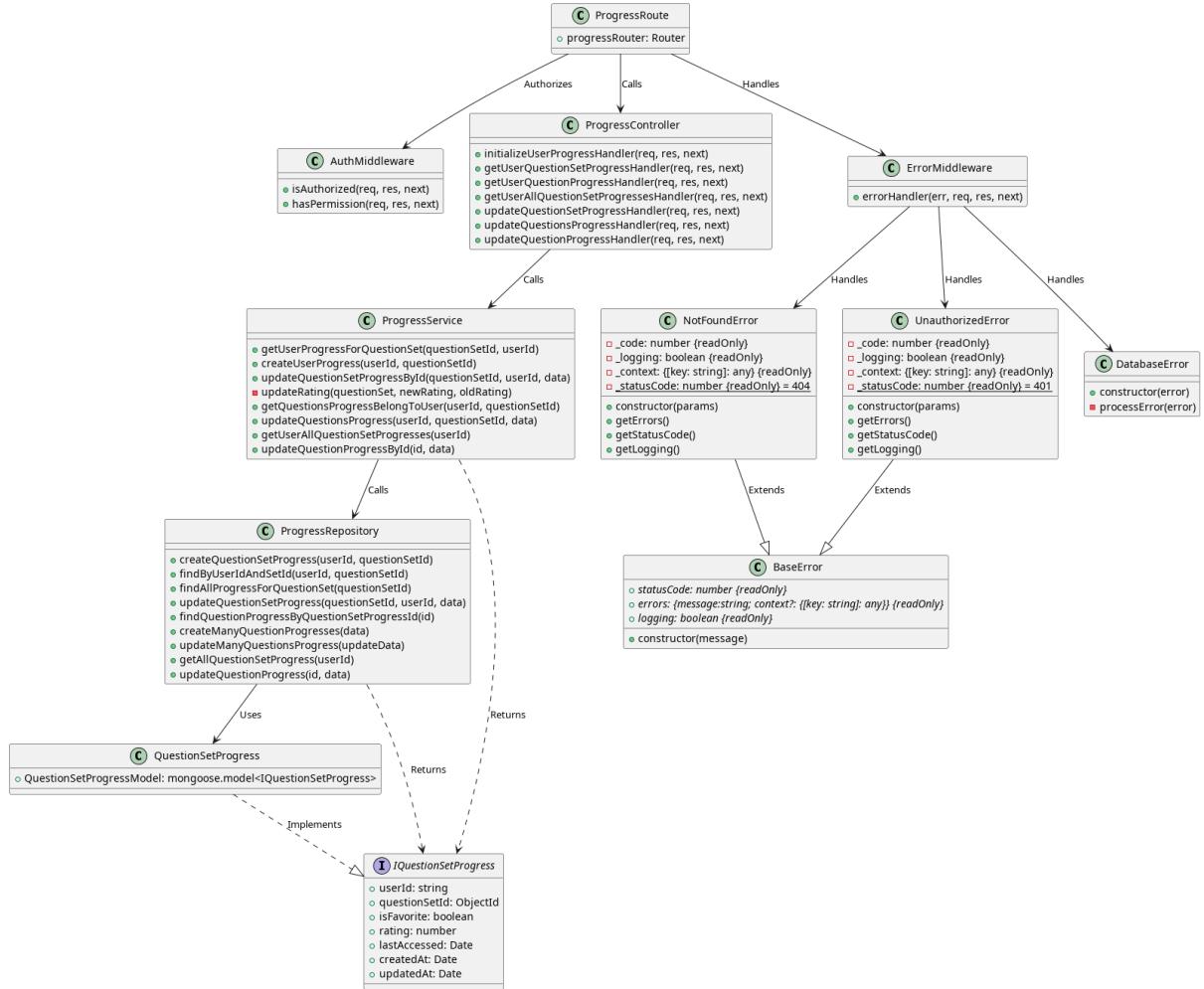


Figure 3.12.1.1: view progress¹⁸⁵

¹⁸⁵ [view progress.png](#)

3.12.1.2 Sequence diagram

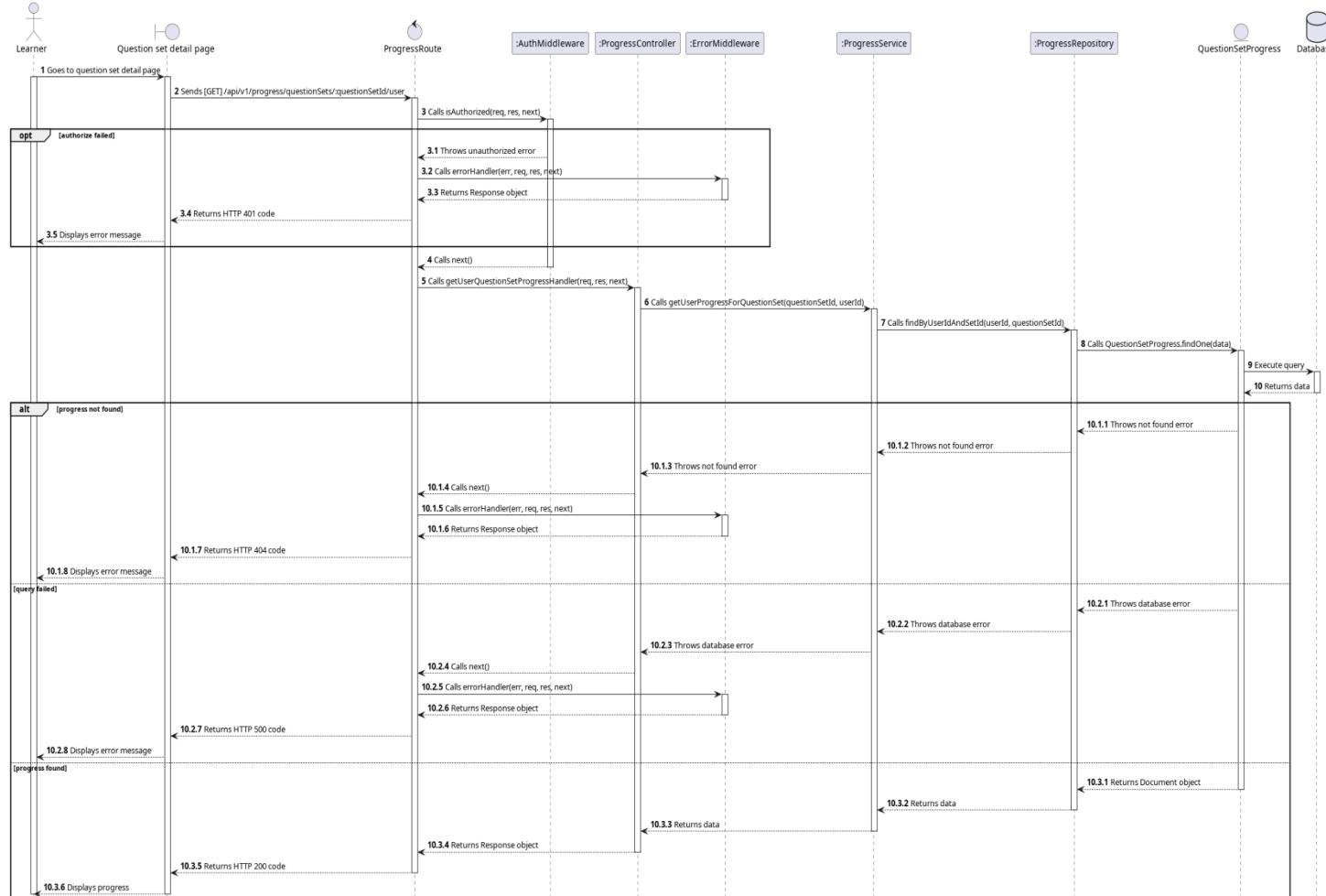


Figure 3.12.1.2.a: Check progress exist¹⁸⁶

186 [view question progress.png](#)

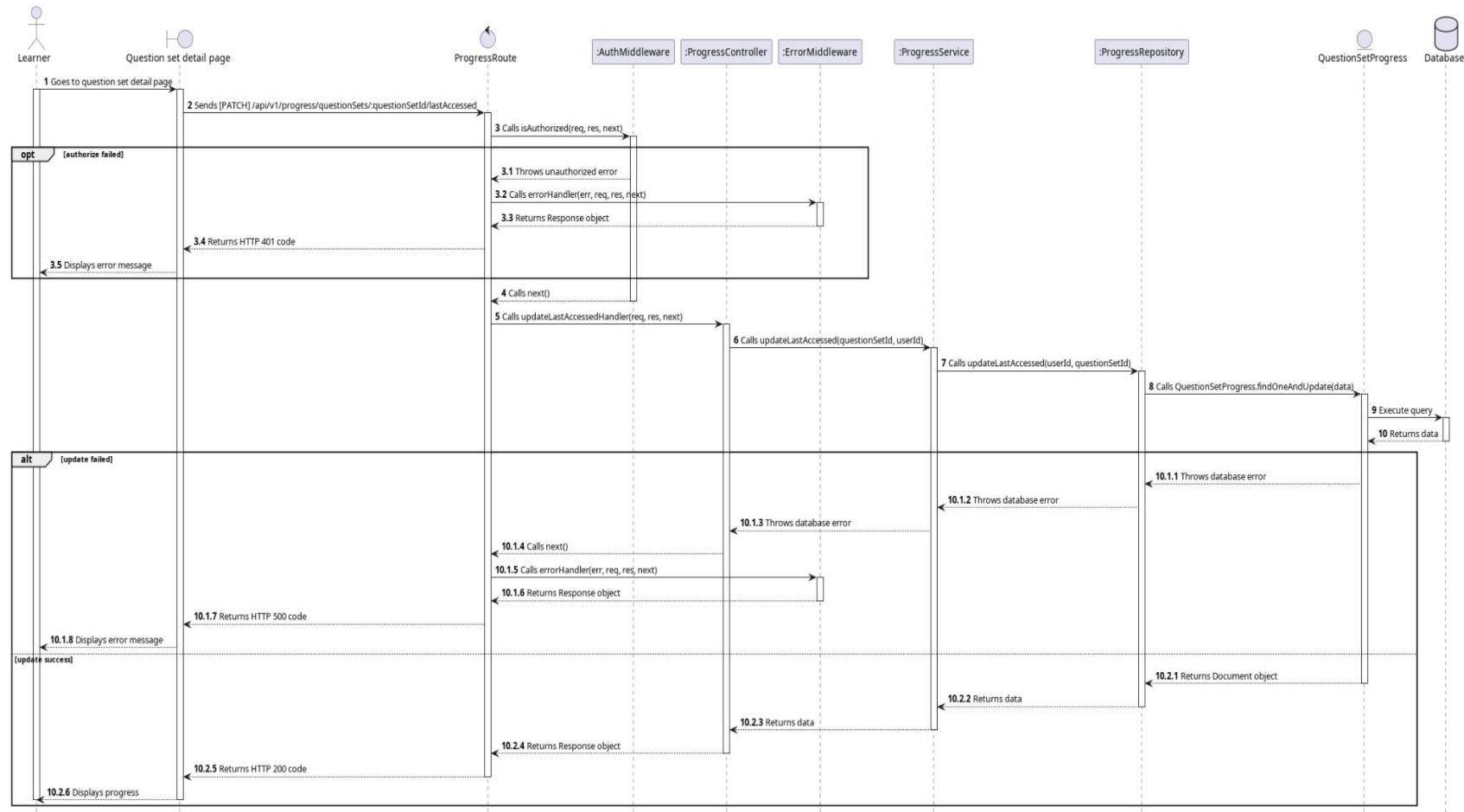


Figure 3.12.1.2.b: Update last access date¹⁸⁷

187 [update from last access.png](#)

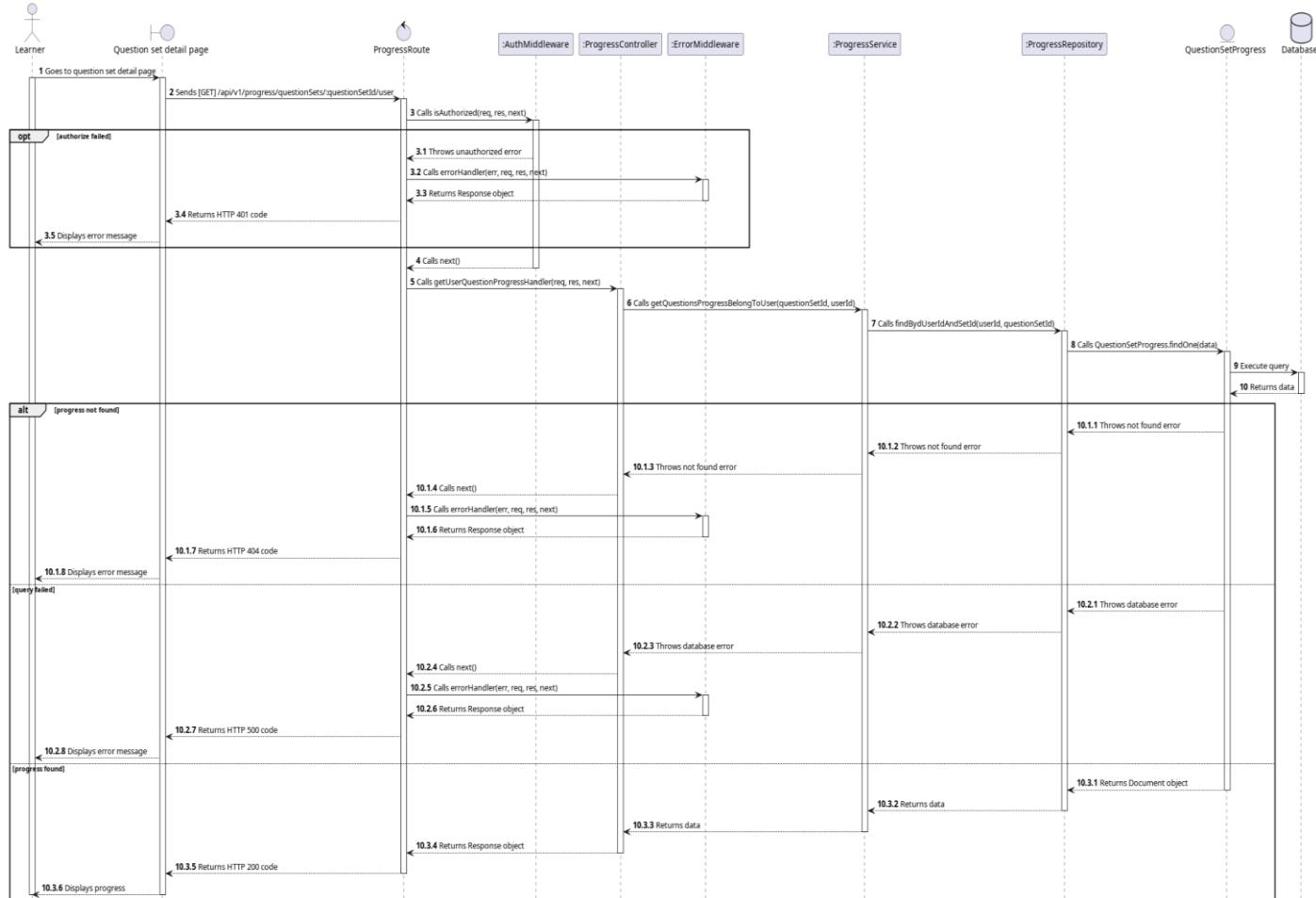


Figure 3.12.1.2.c: Get progress¹⁸⁸

3.12.2 Filter questions

3.12.2.1 Class diagram

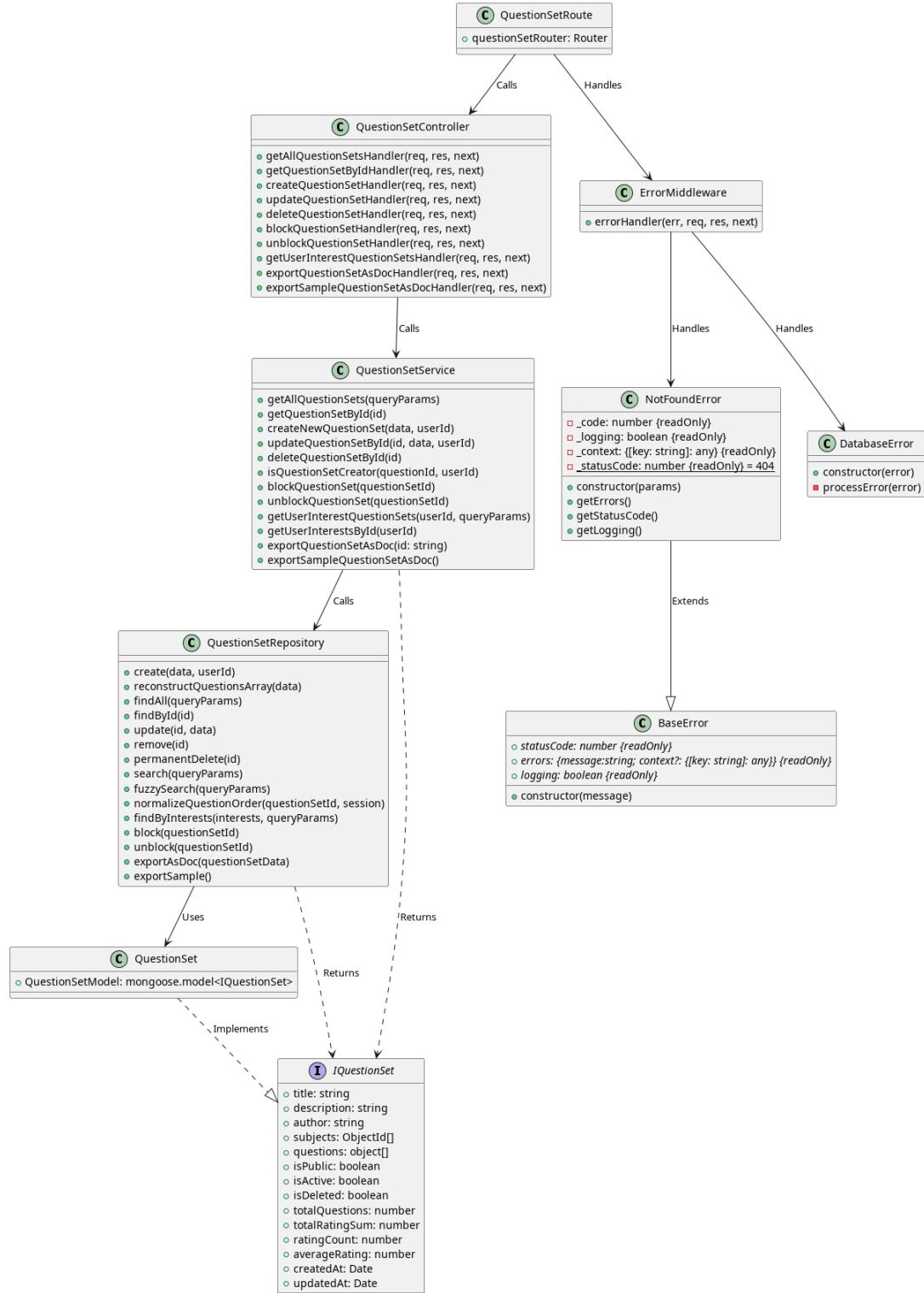


Figure 3.12.2.1: Filter questions¹⁸⁹

189 [filter questions.png](#)

3.12.2.2 Sequence diagram

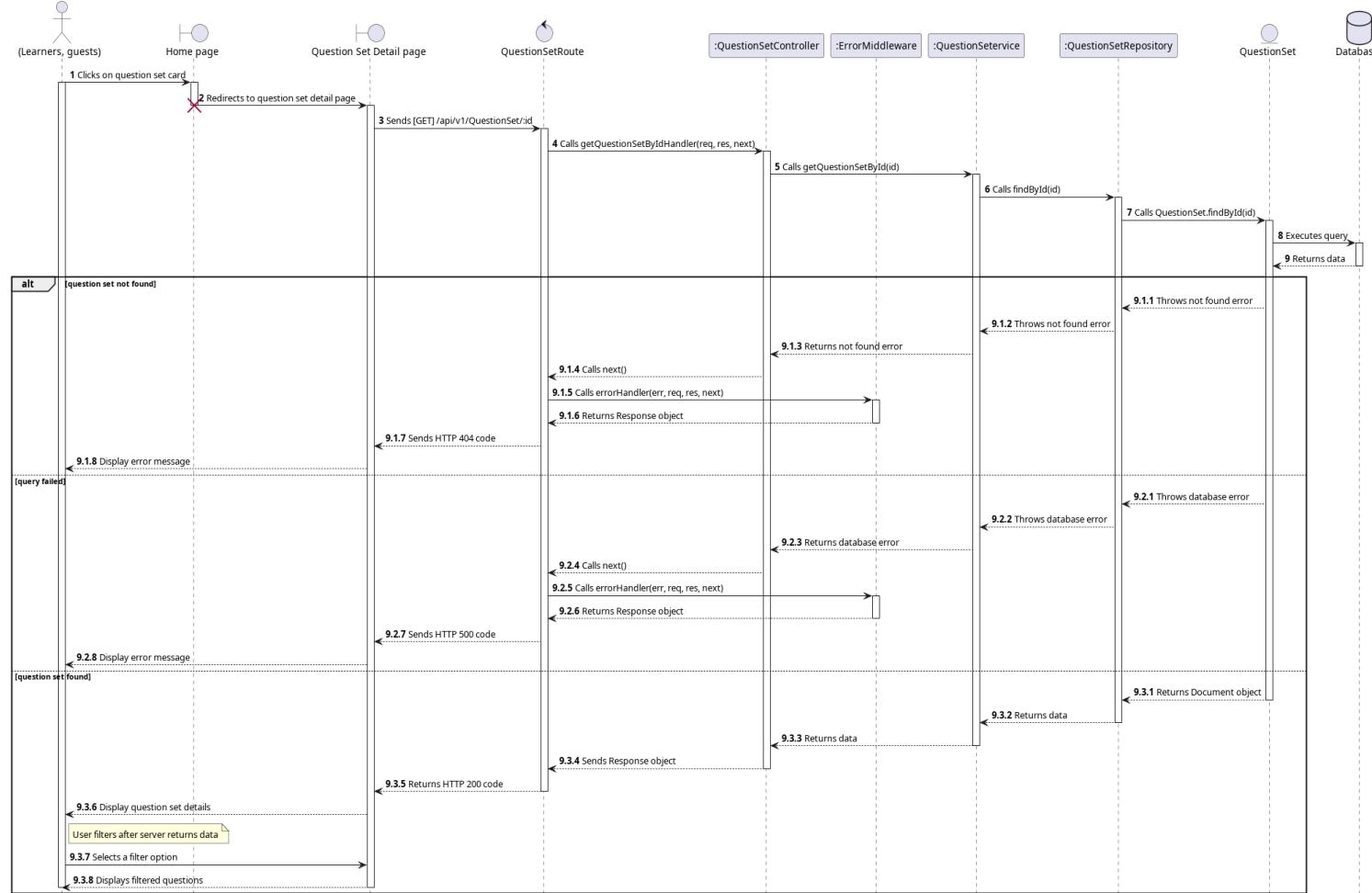


Figure 3.12.2.2: Filter questions¹⁹⁰

190 [filter questions.png](#)

3.12.3 Mark questions

3.12.3.1 Class diagram

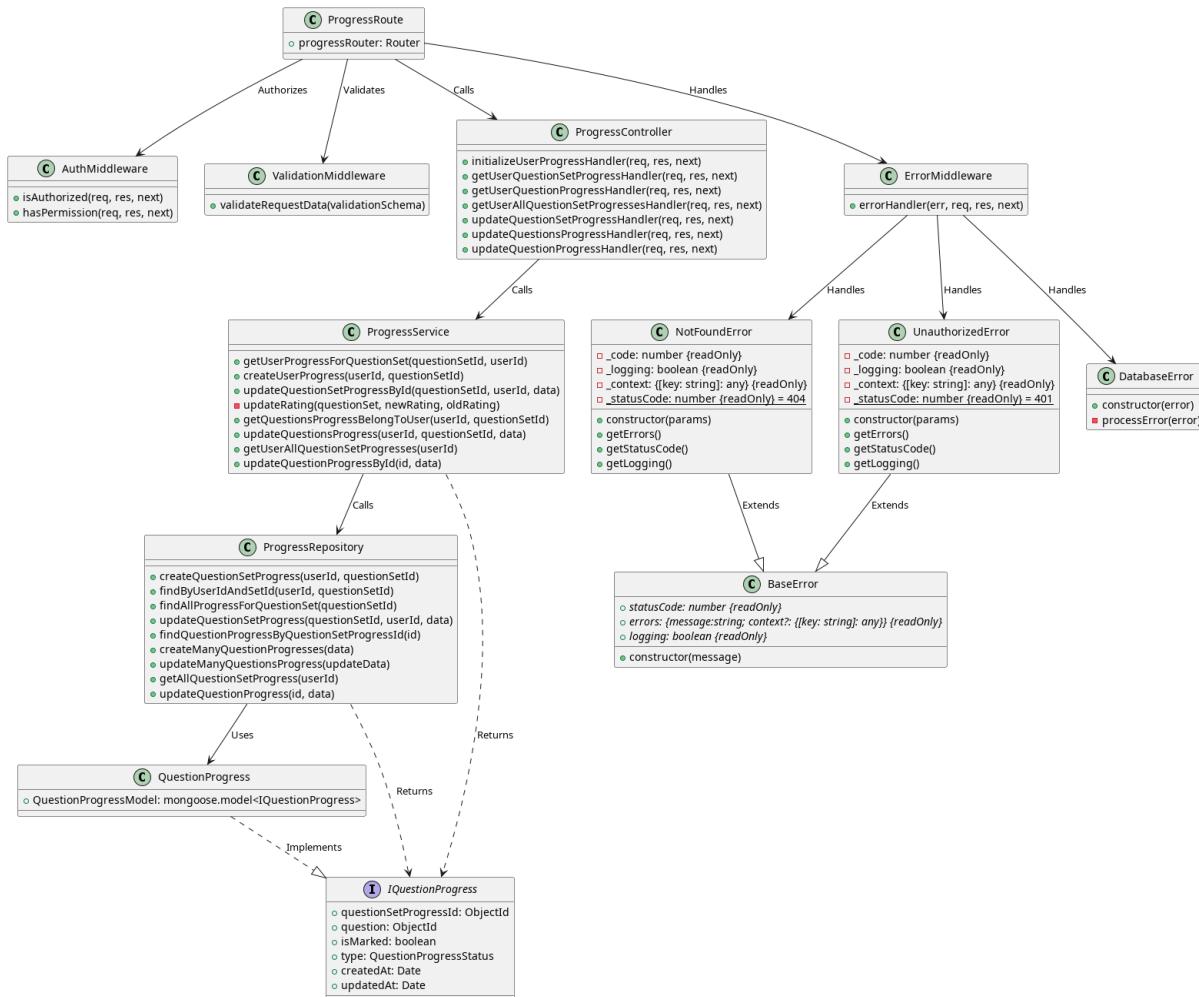


Figure 3.12.3.1: Mark questions¹⁹¹

¹⁹¹ mark question.png

3.12.3.2 Sequence diagram

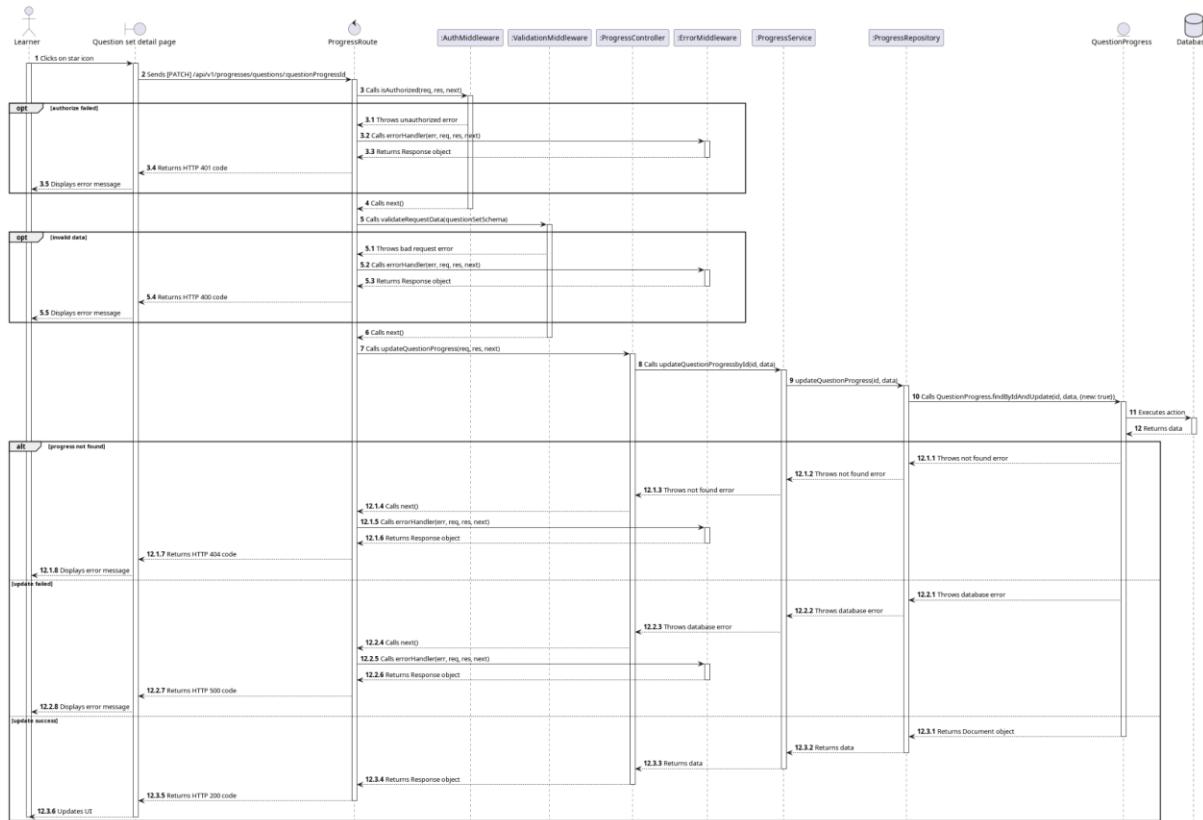


Figure 3.12.3.2: Mark questions¹⁹²

¹⁹² mark question.png

3.13 Comment Management

3.13.1 View comments

3.13.1.1 Class diagram

3.13.1.1.1 View comments

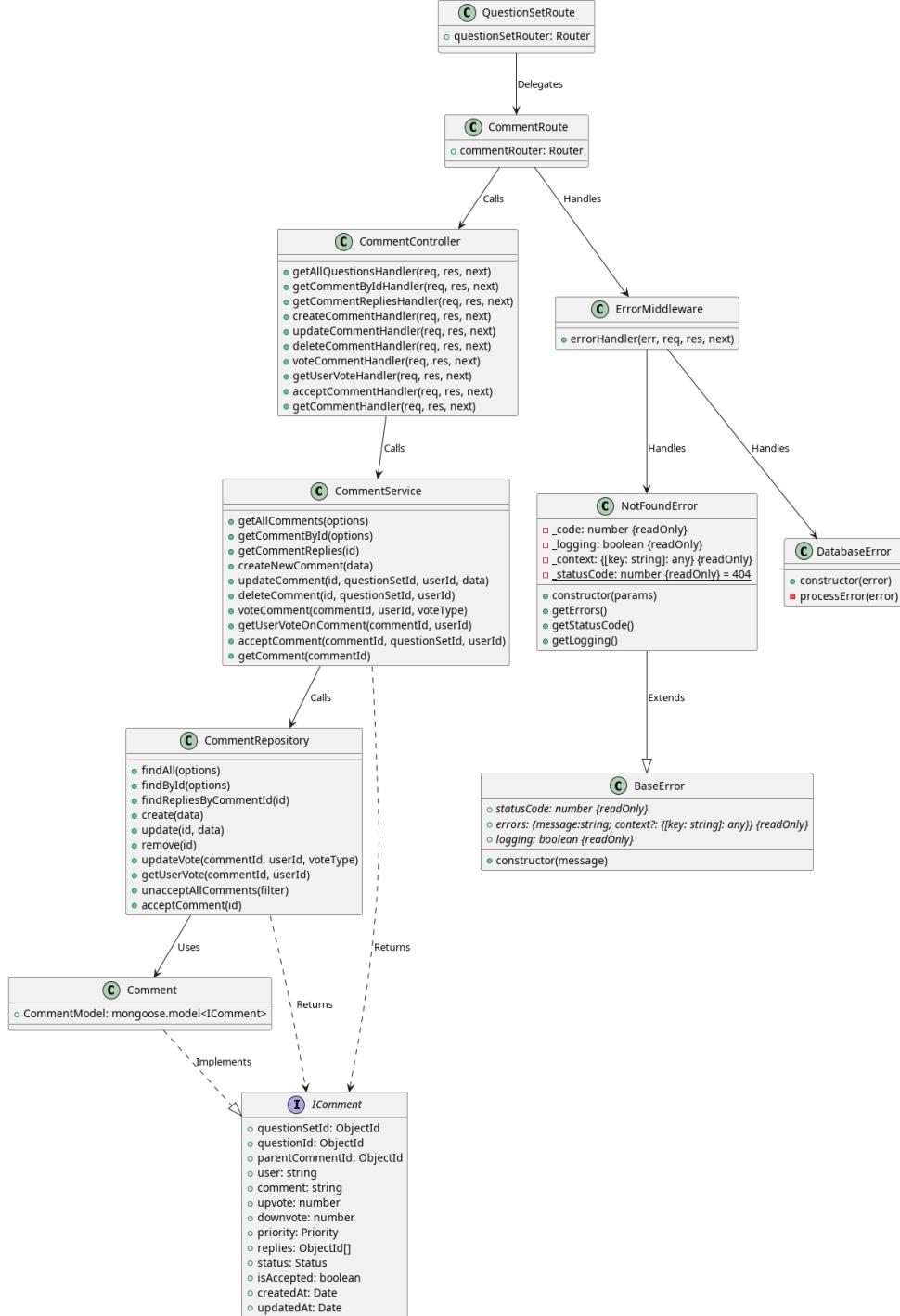


Figure 3.13.1.1.1: View comments¹⁹³

¹⁹³ [view question set comments.png](#)

3.13.1.1.2 View reply comments

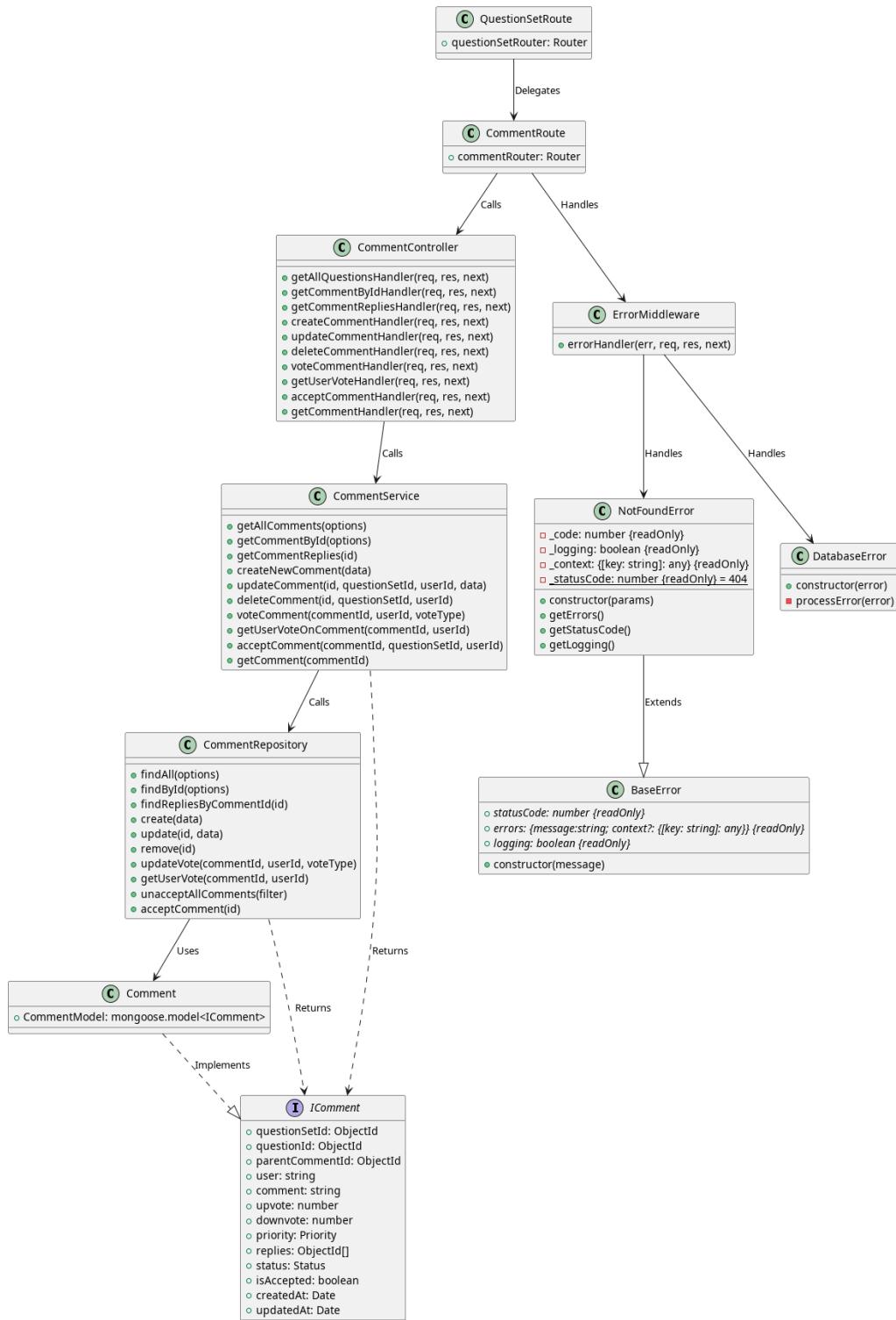


Figure 3.13.1.1.2: View reply comments¹⁹⁴

¹⁹⁴ [view question set comments.png](#)

3.13.1.2 Sequence diagram

3.13.1.2.1 View comments

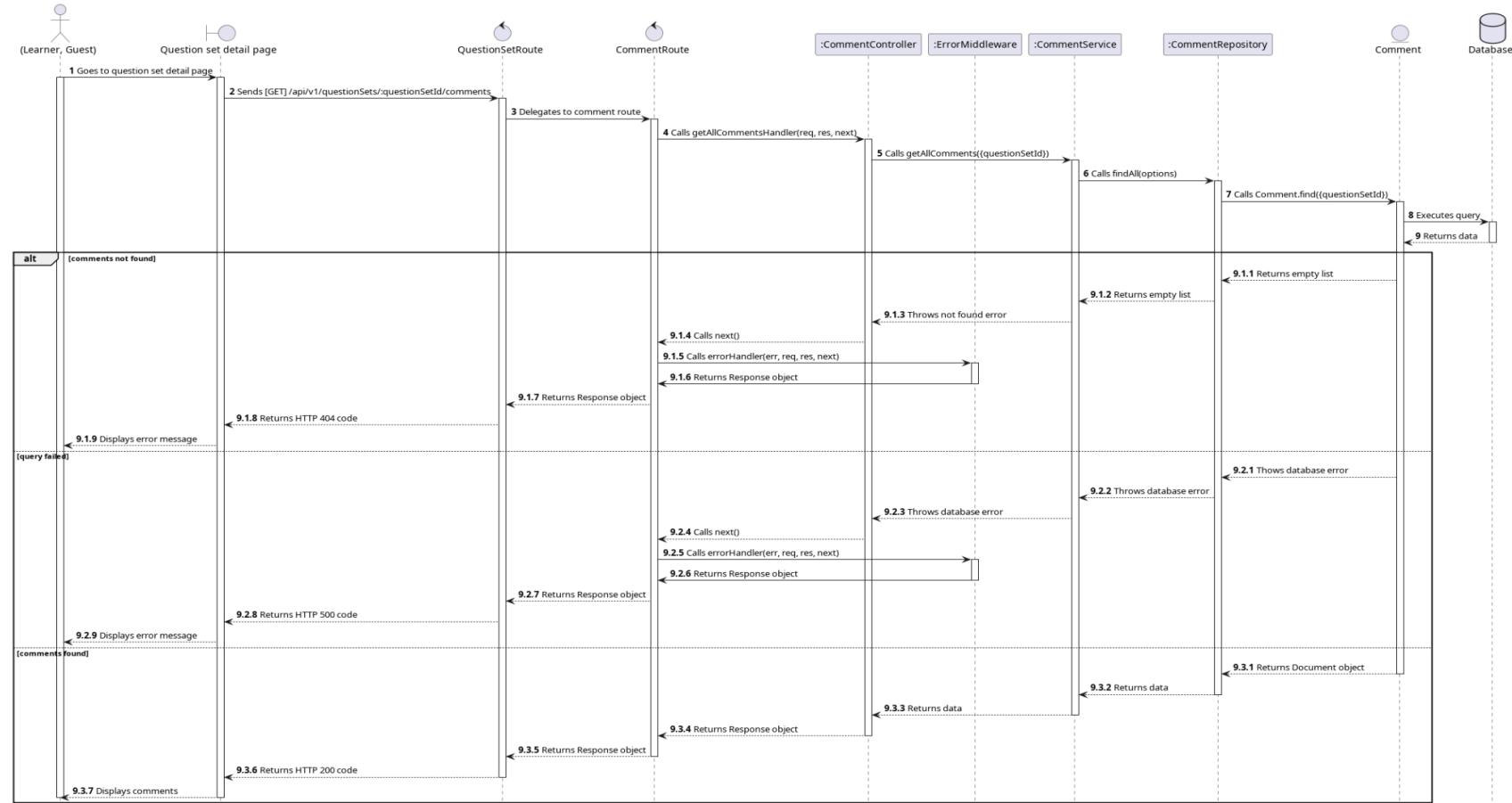


Figure 3.13.1.2.1: View comments¹⁹⁵

¹⁹⁵ [view question set comments.png](#)

3.13.1.2.2 View reply comments

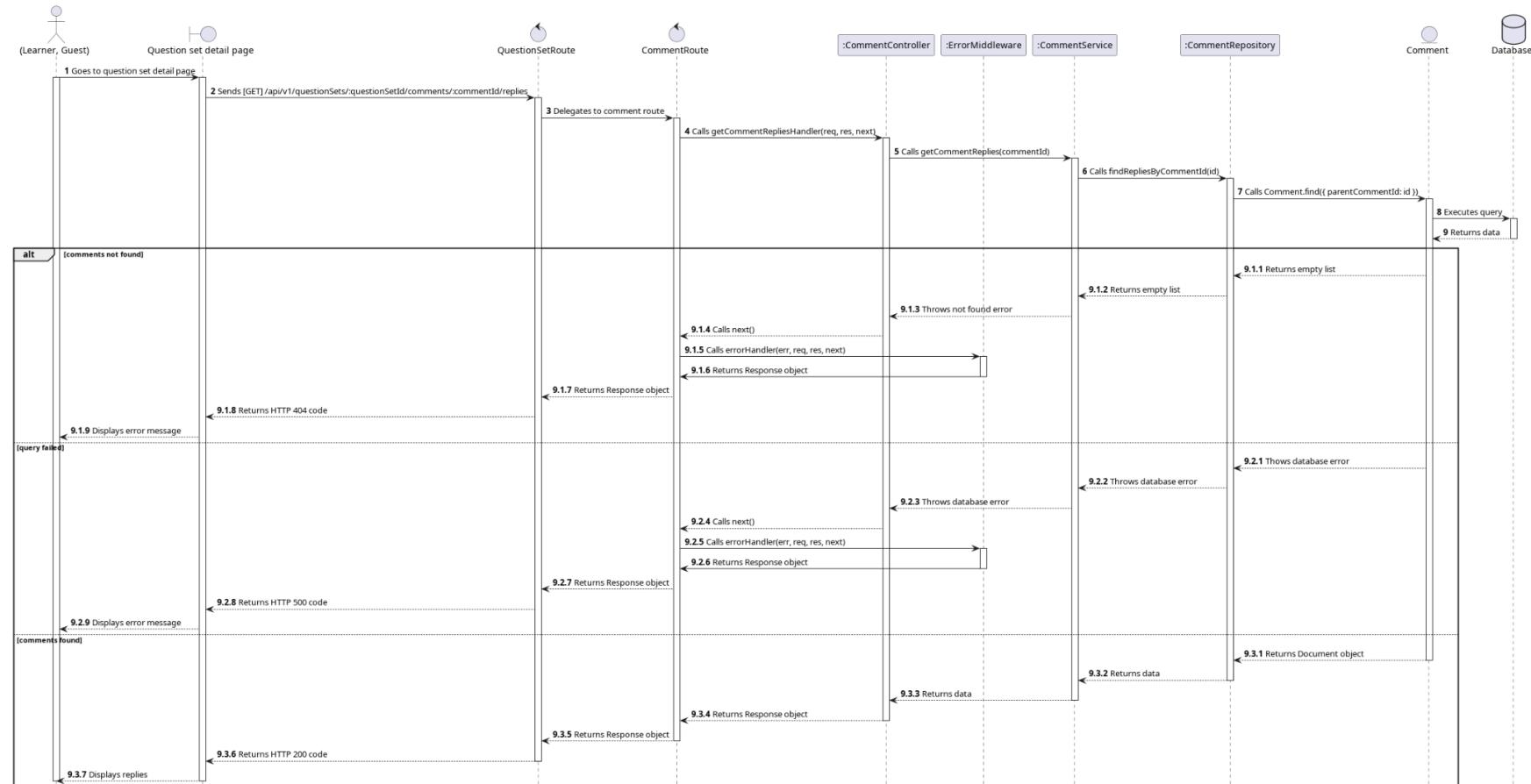


Figure 3.13.1.2.2: View reply comments¹⁹⁶

¹⁹⁶ [view comments replies.png](#)

3.13.2 Create comments

3.13.2.1 Class diagram

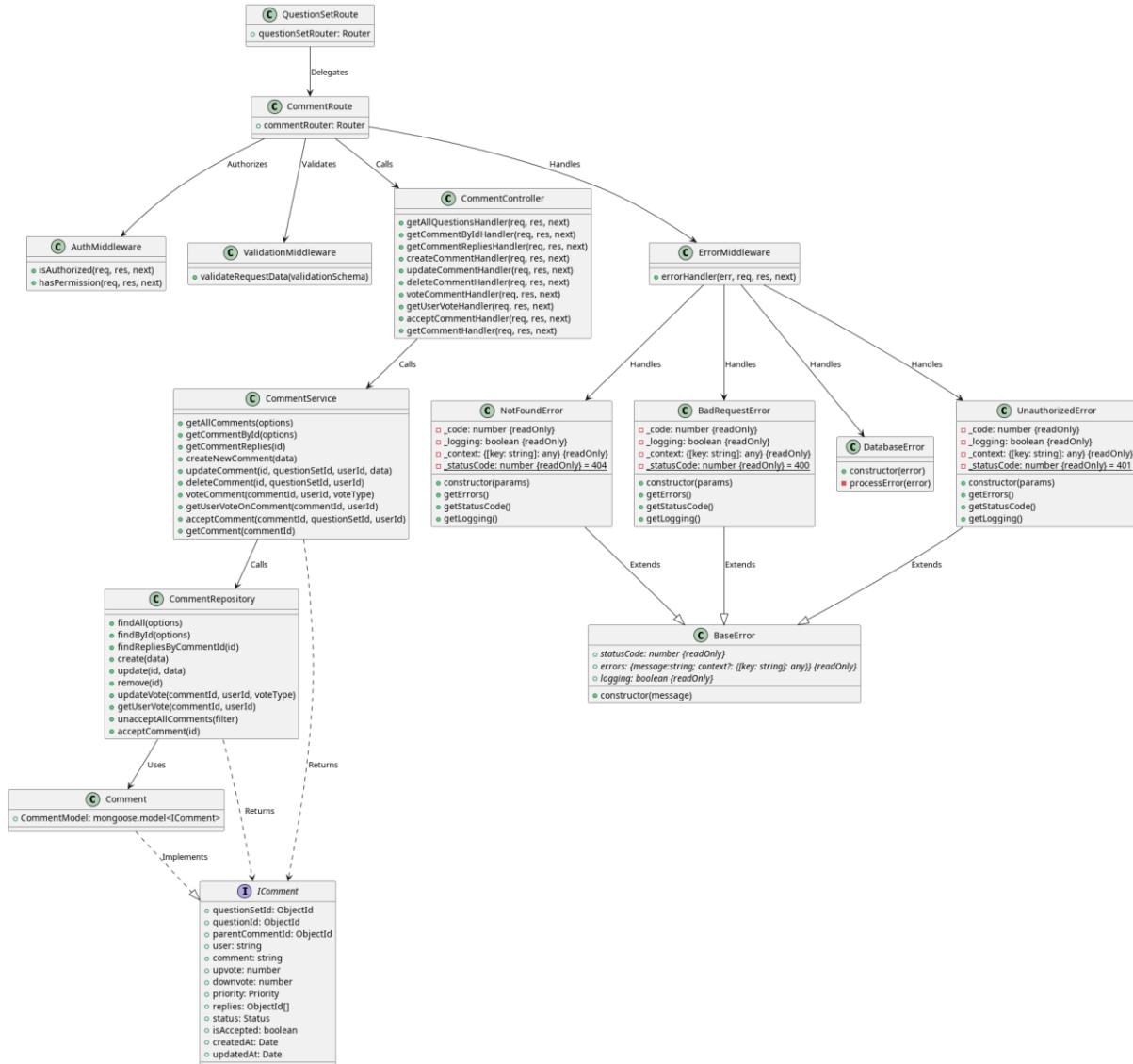


Figure 3.13.2.1: Create comments¹⁹⁷

¹⁹⁷ [create comment.png](#)

3.13.2.2 Sequence diagram

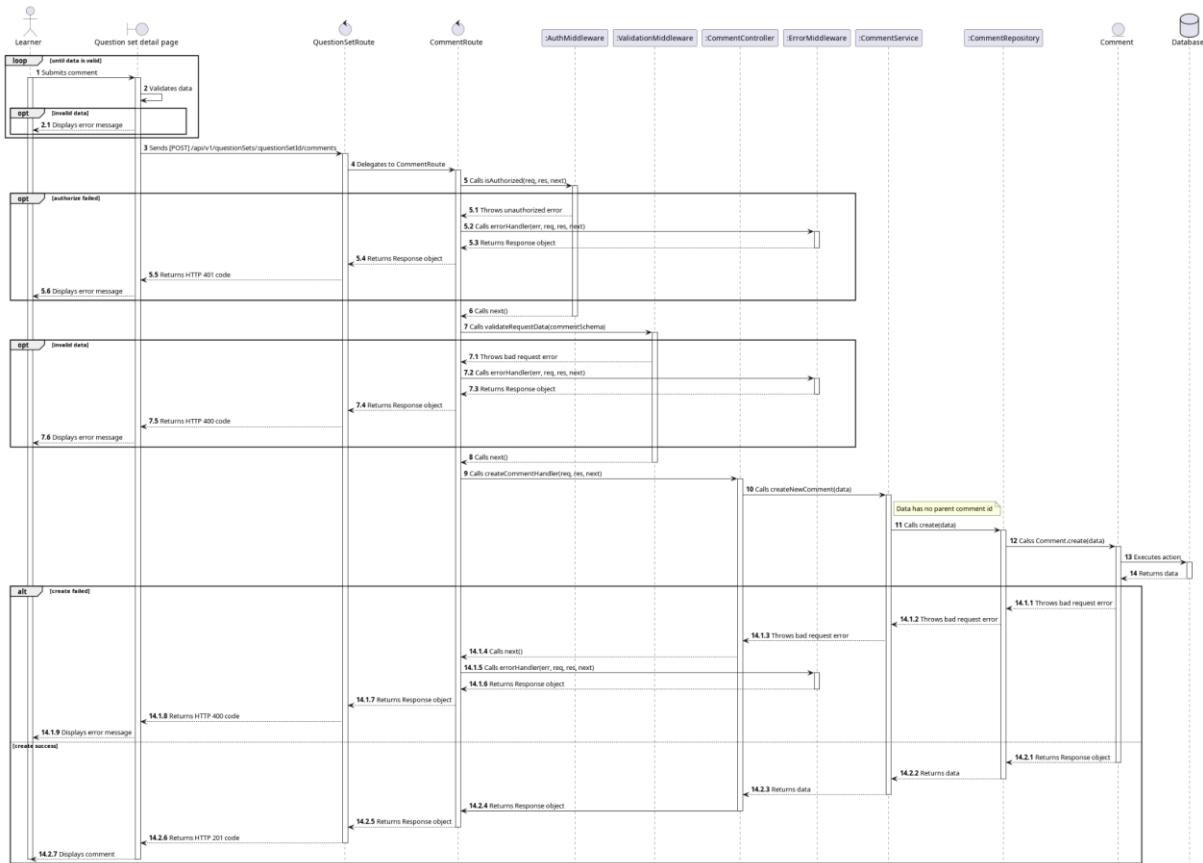


Figure 3.13.2.2: Create comments¹⁹⁸

198 [create comments.png](#)

3.13.3 Reply comments

3.13.3.1 Class diagram

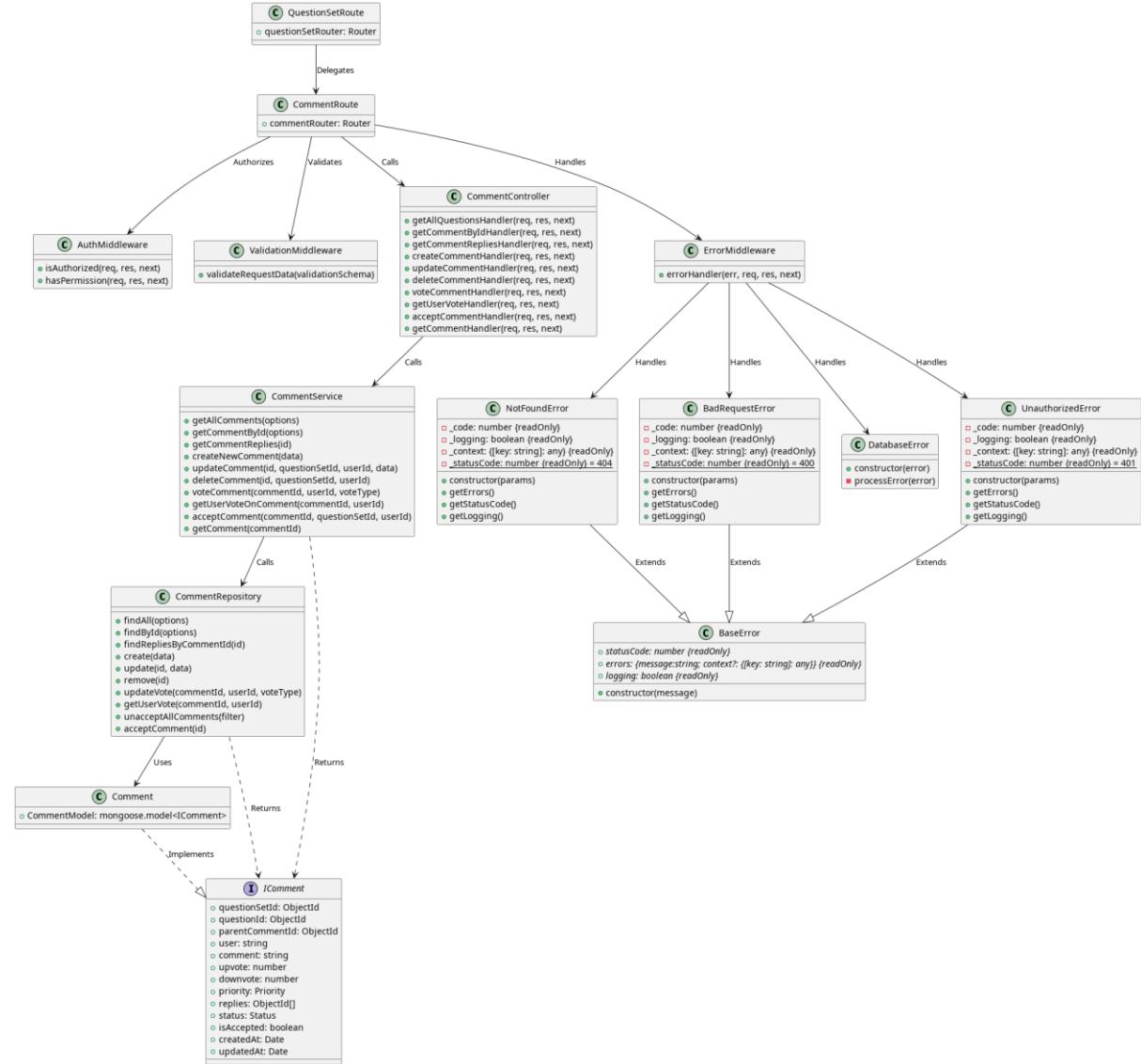


Figure 3.13.3.1: Reply comments¹⁹⁹

¹⁹⁹ [create comment.png](#)

3.13.3.2 Sequence diagram

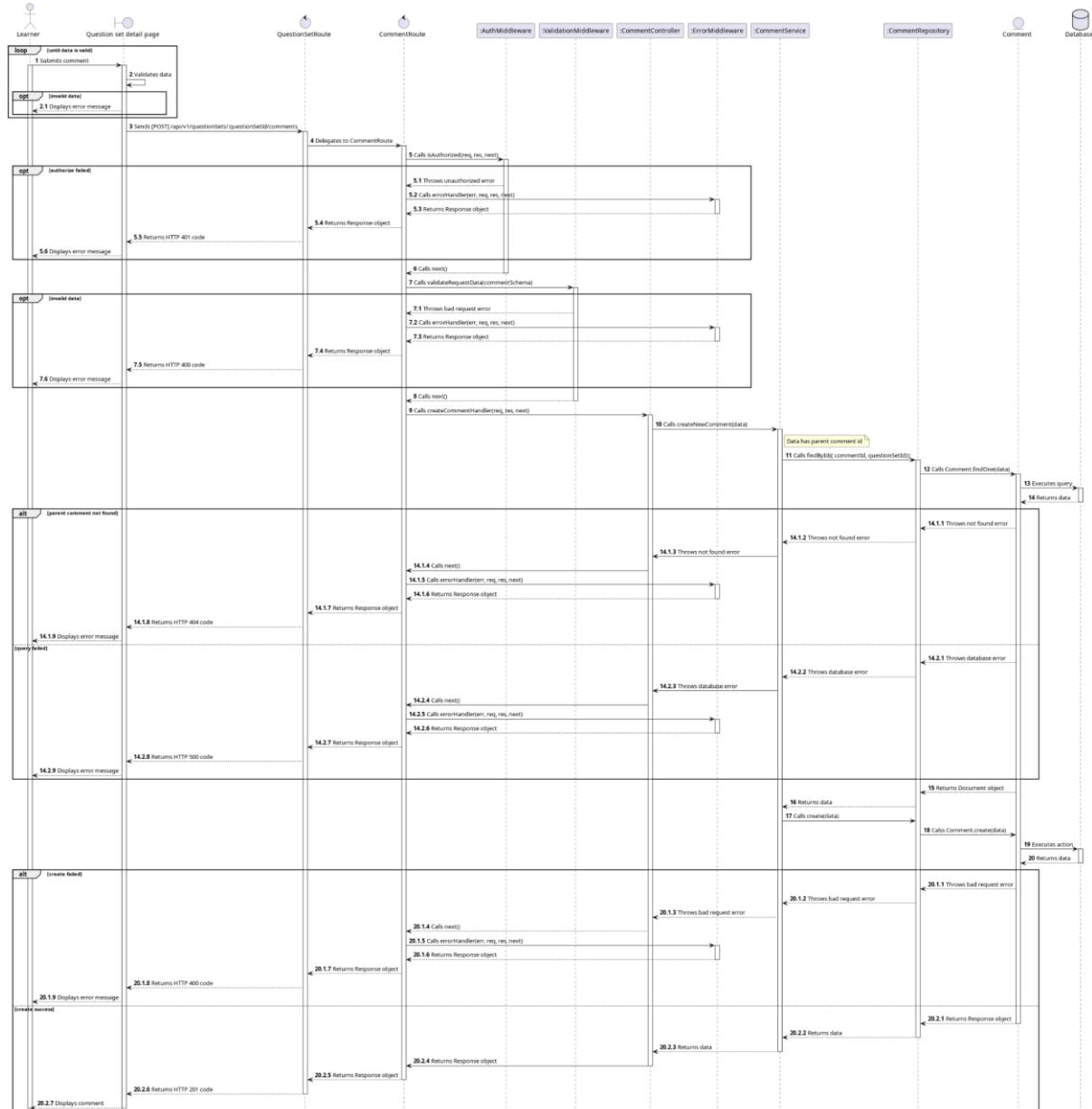


Figure 3.13.3.2: Reply comments²⁰⁰

²⁰⁰ create comments.png

3.13.4 Delete comments

3.13.4.1 Class diagram

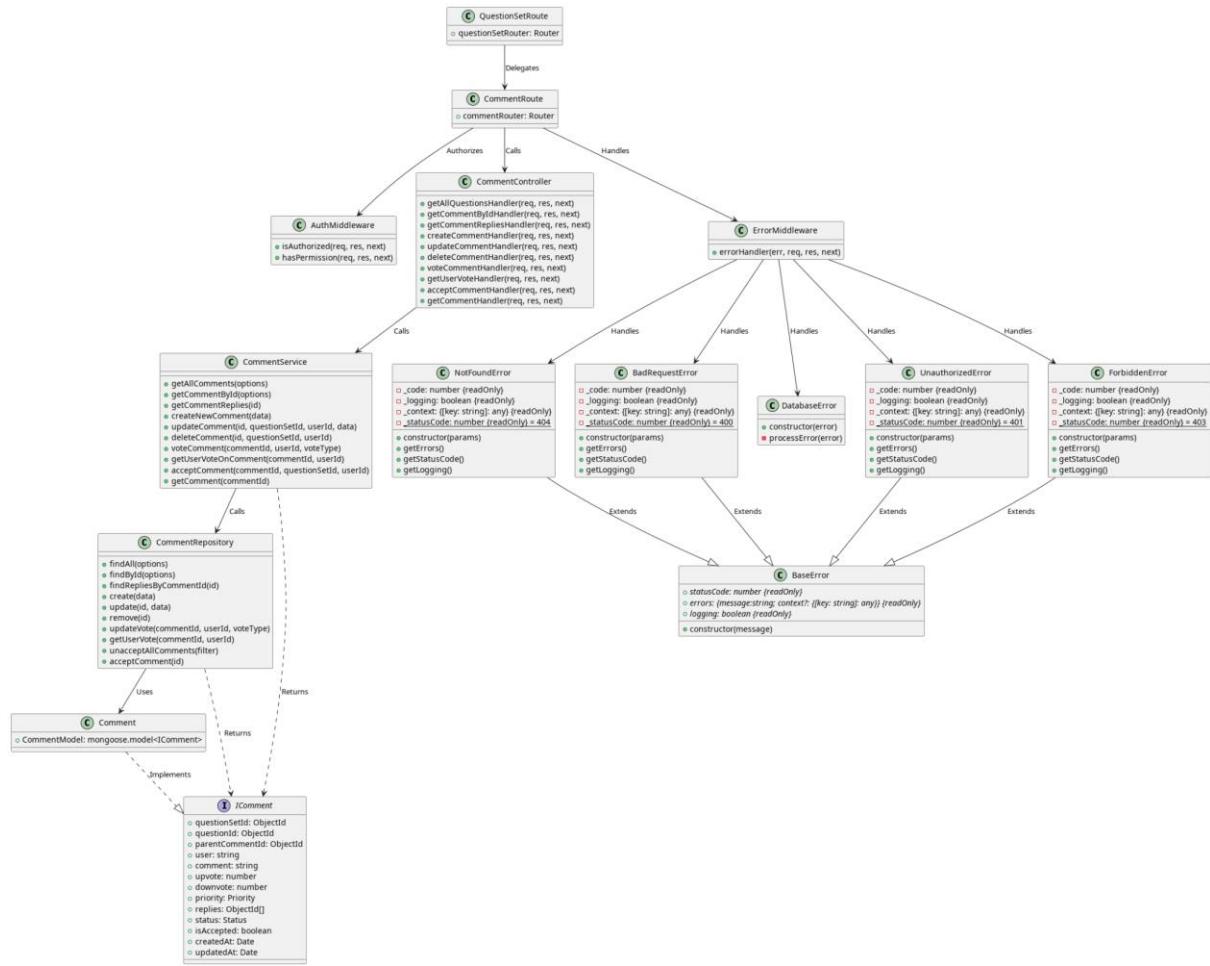


Figure 3.13.4.1: Delete comments²⁰¹

²⁰¹ [update comment.png](#)

3.13.4.2 Sequence diagram

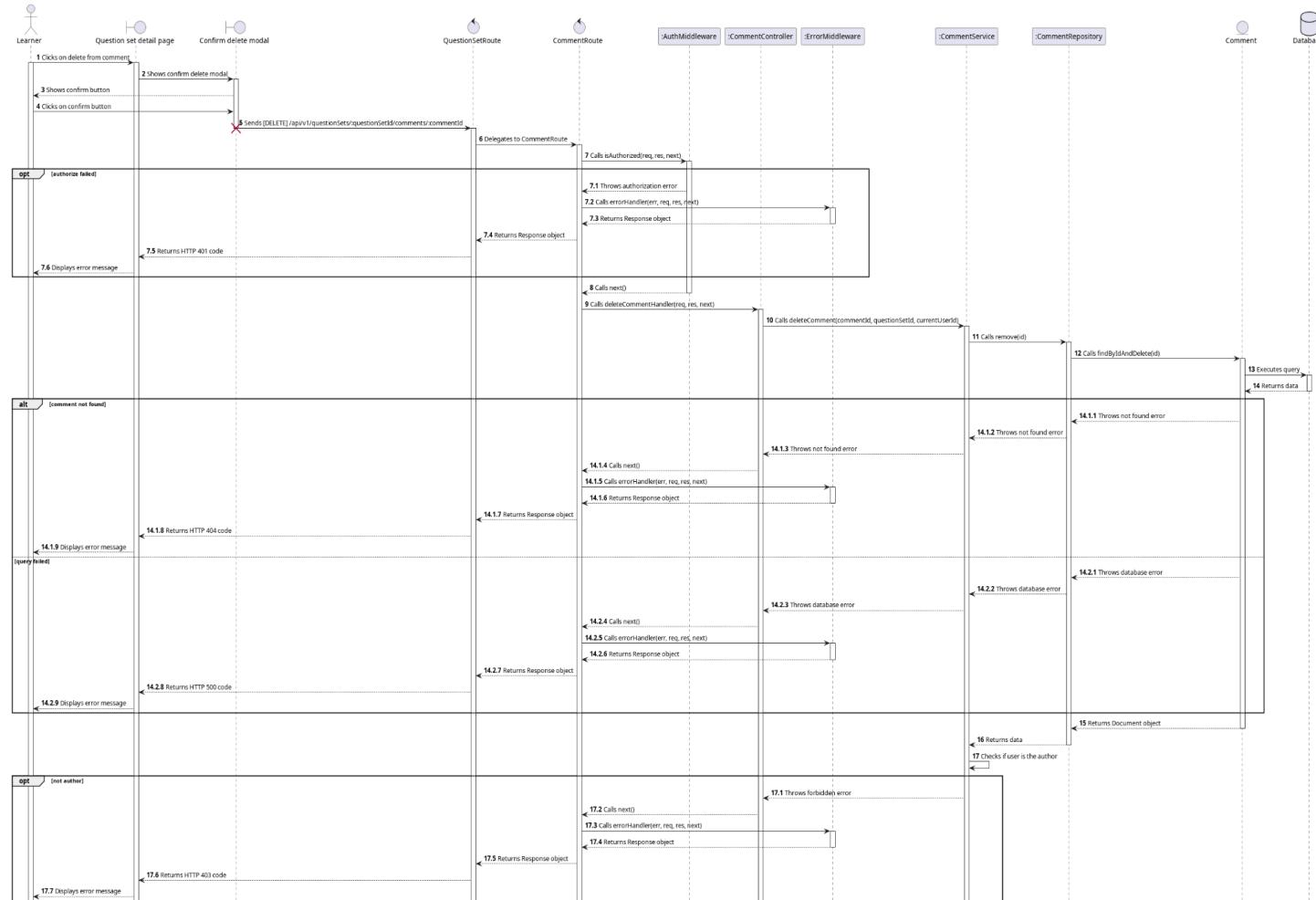


Figure 3.13.4.2.a: Delete comments - 1²⁰²

202 [delete comment.png](#)

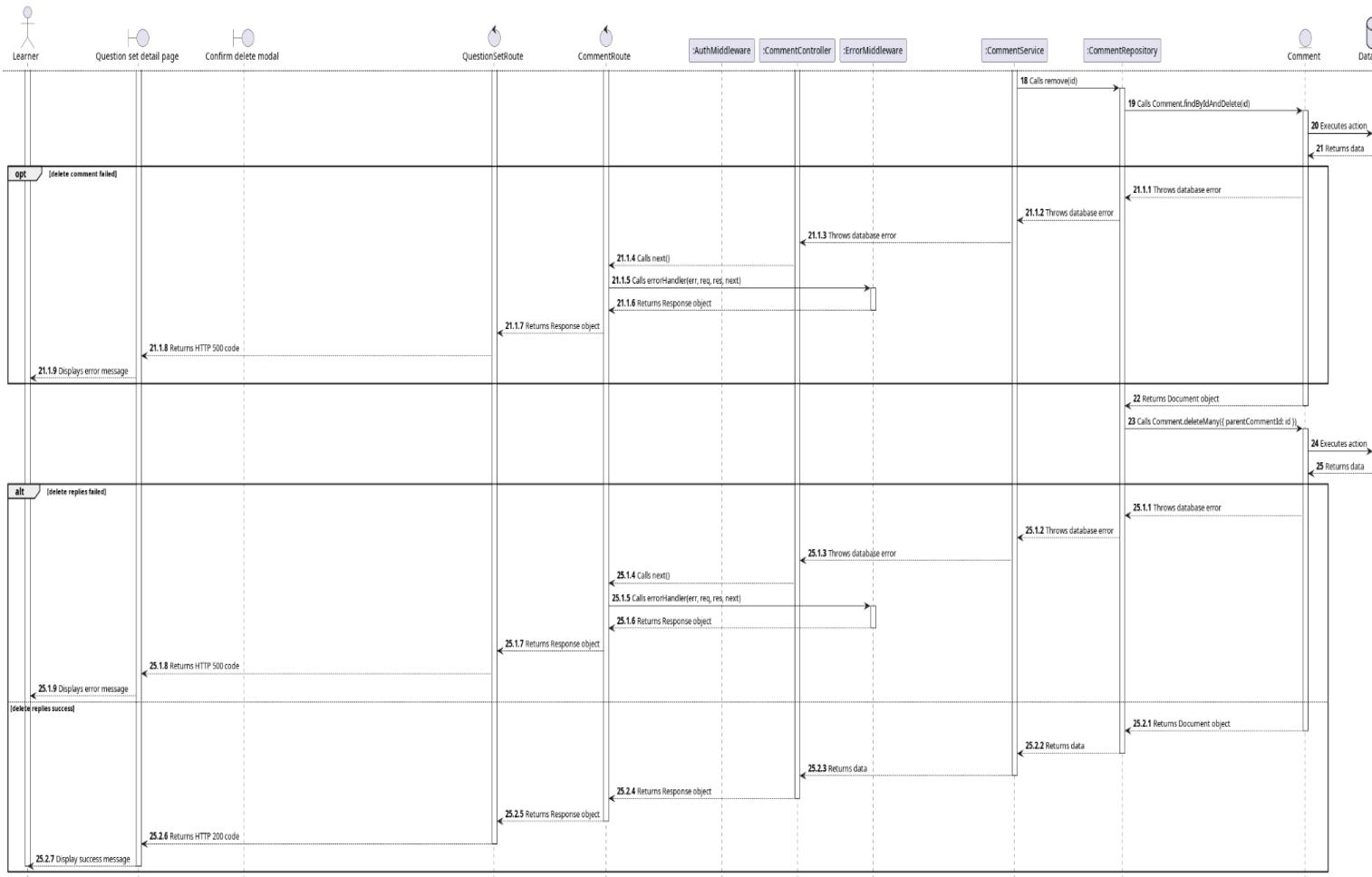


Figure 3.13.4.2.b: Delete comments - 2

3.13.5 Upvote comments

3.13.5.1 Class diagram

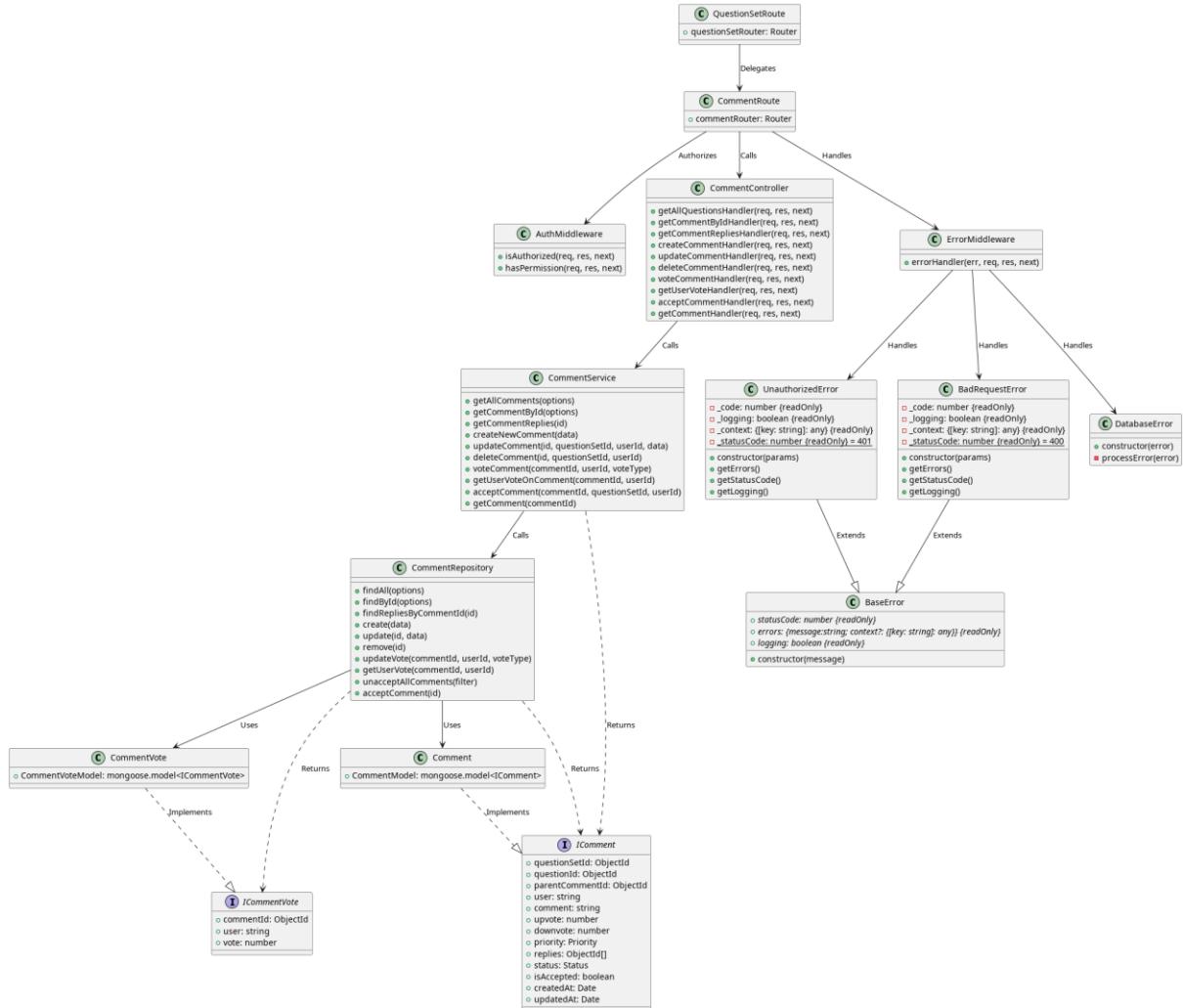


Figure 3.13.5.1: Upvote comments²⁰³

203 upvote comment.png

3.13.5.2 Sequence diagram

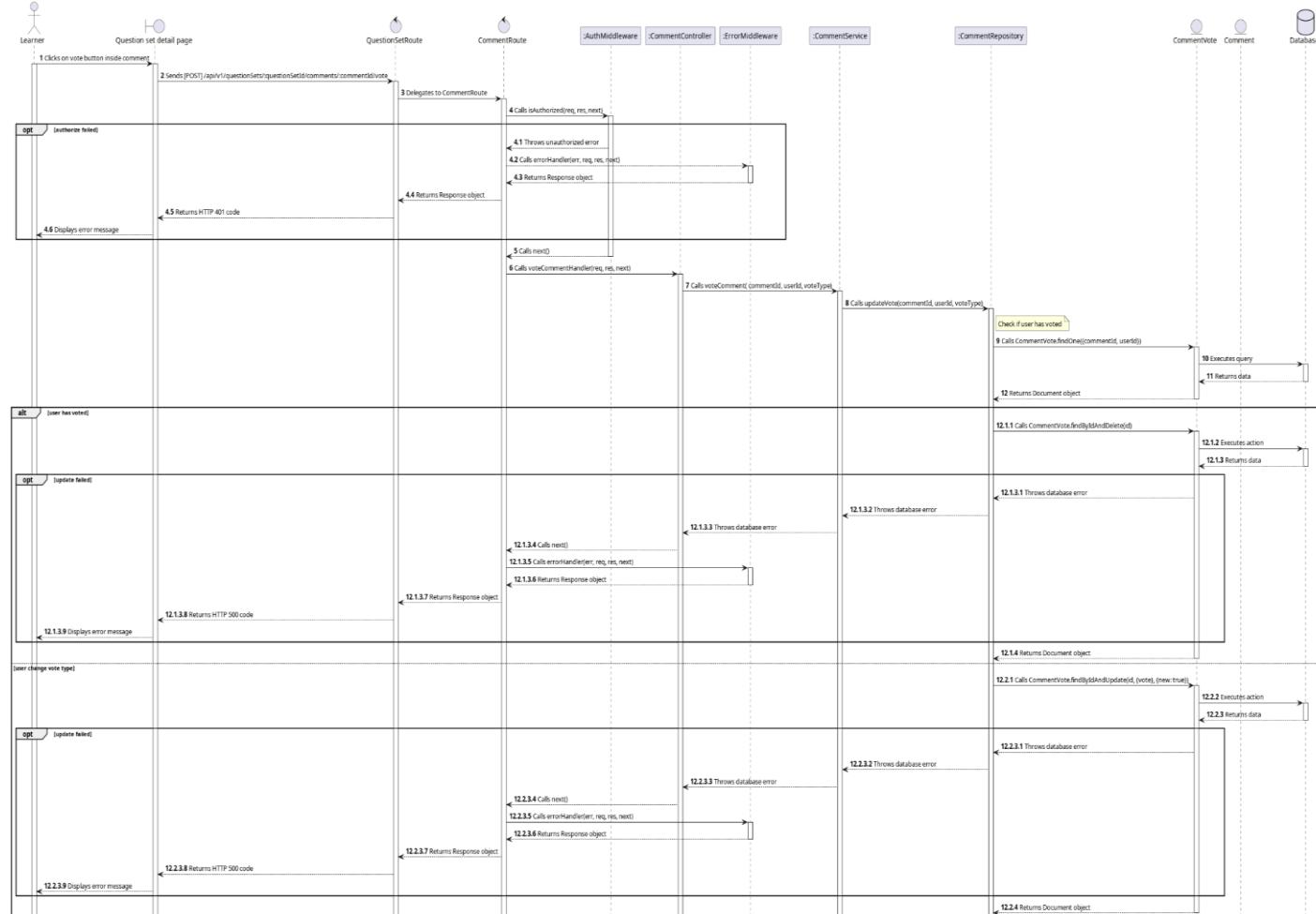


Figure 3.13.5.2.a: Upvote comments - 1²⁰⁴

204 [upvote comment.png](#)

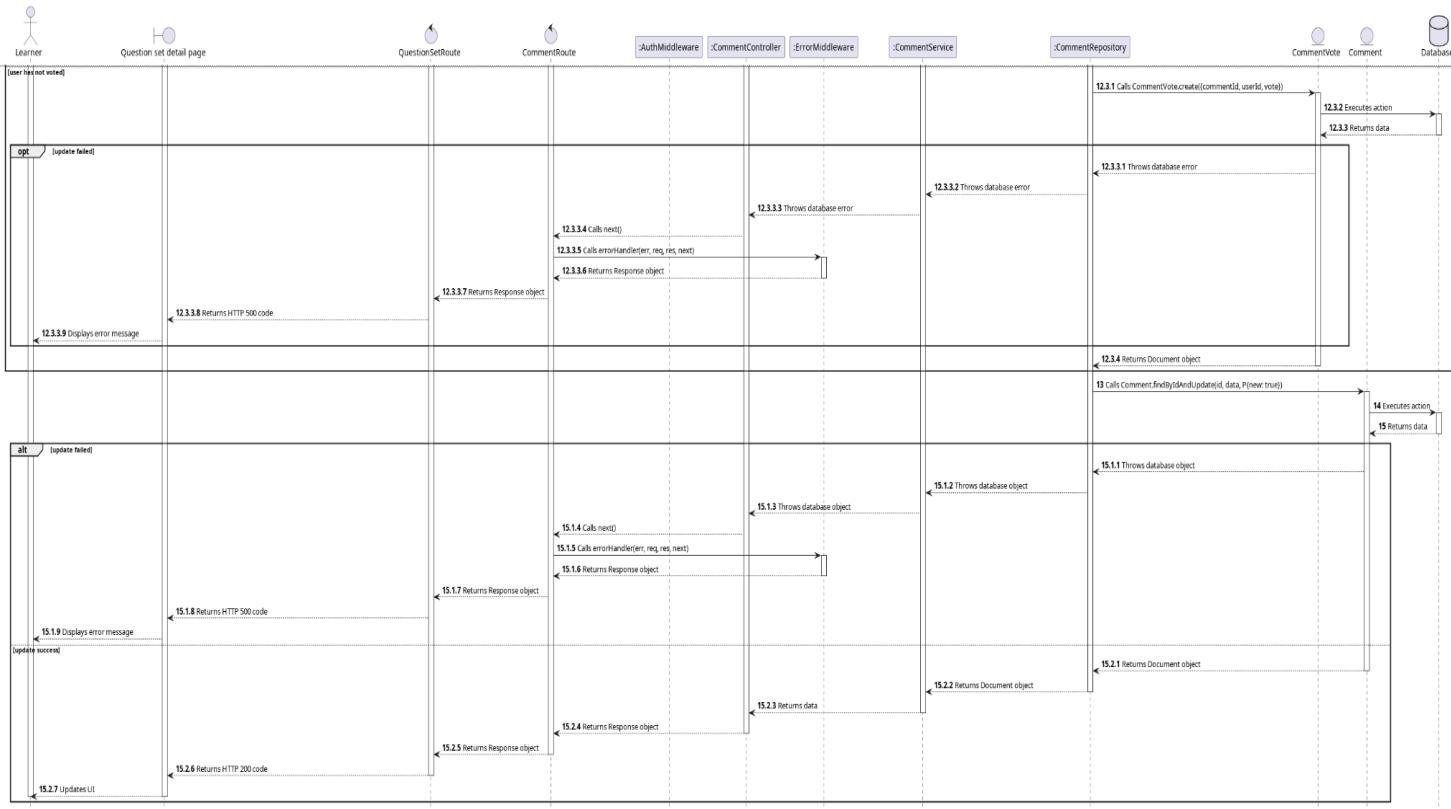


Figure 3.13.5.2.b: Upvote comments - 2

3.13.6 Update comment

3.13.6.1 Class diagram

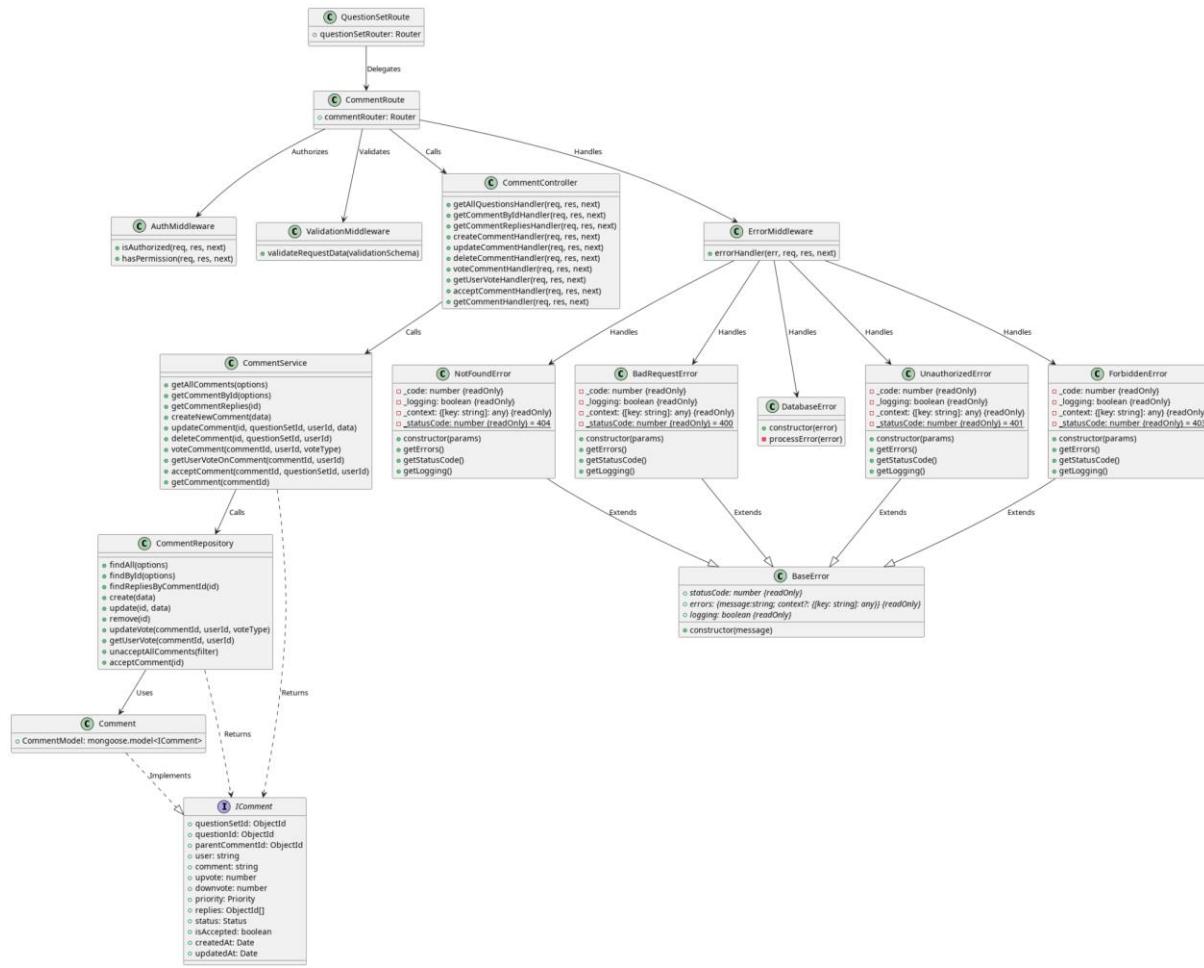


Figure 3.13.6.1: Update comment²⁰⁵

²⁰⁵ [update comment.png](#)

3.13.6.2 Sequence diagram

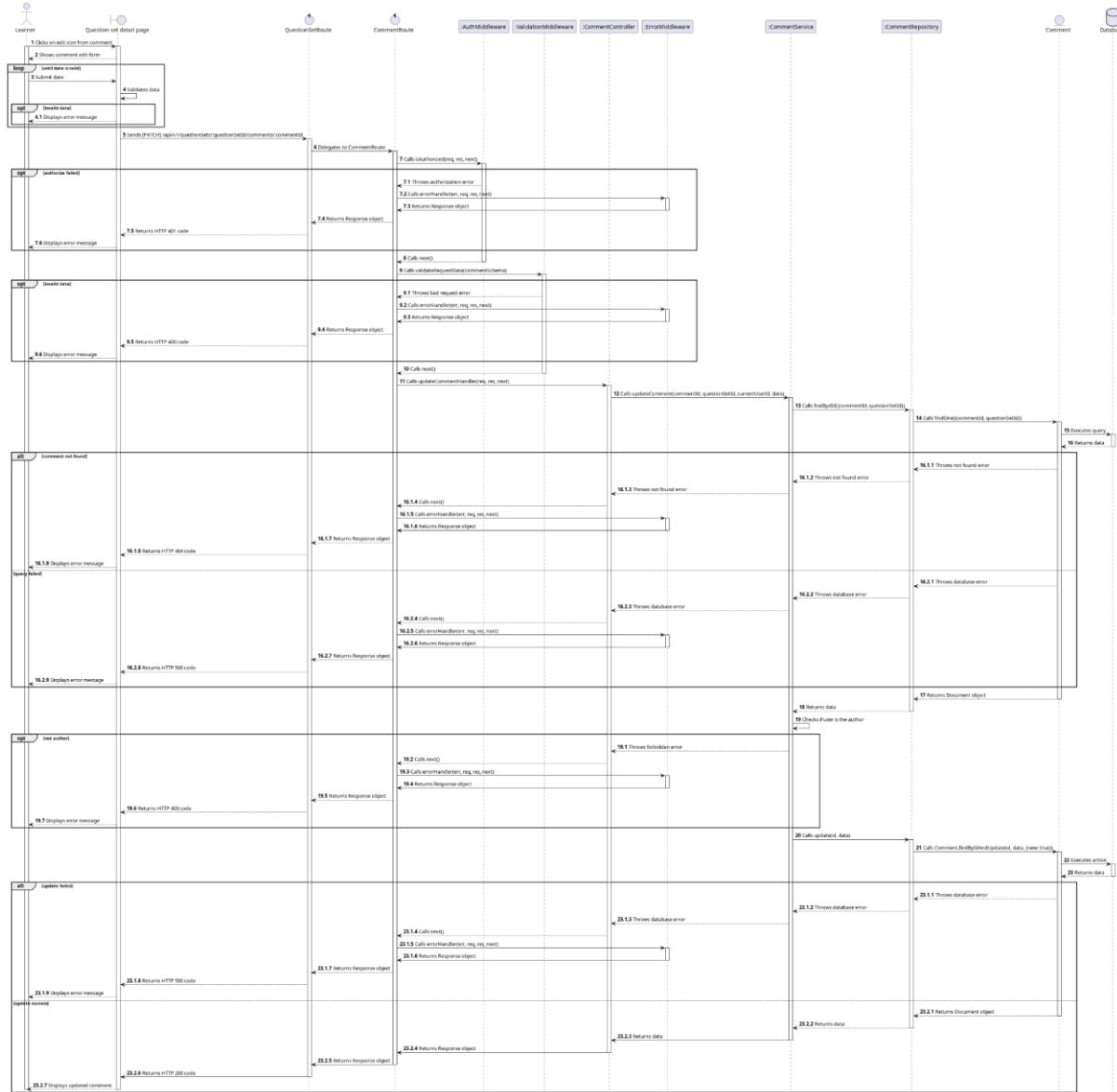


Figure 3.14.6.2: Update comment²⁰⁶

206 update comment.png

3.13.7 Filter comments

3.13.7.1 Class diagram

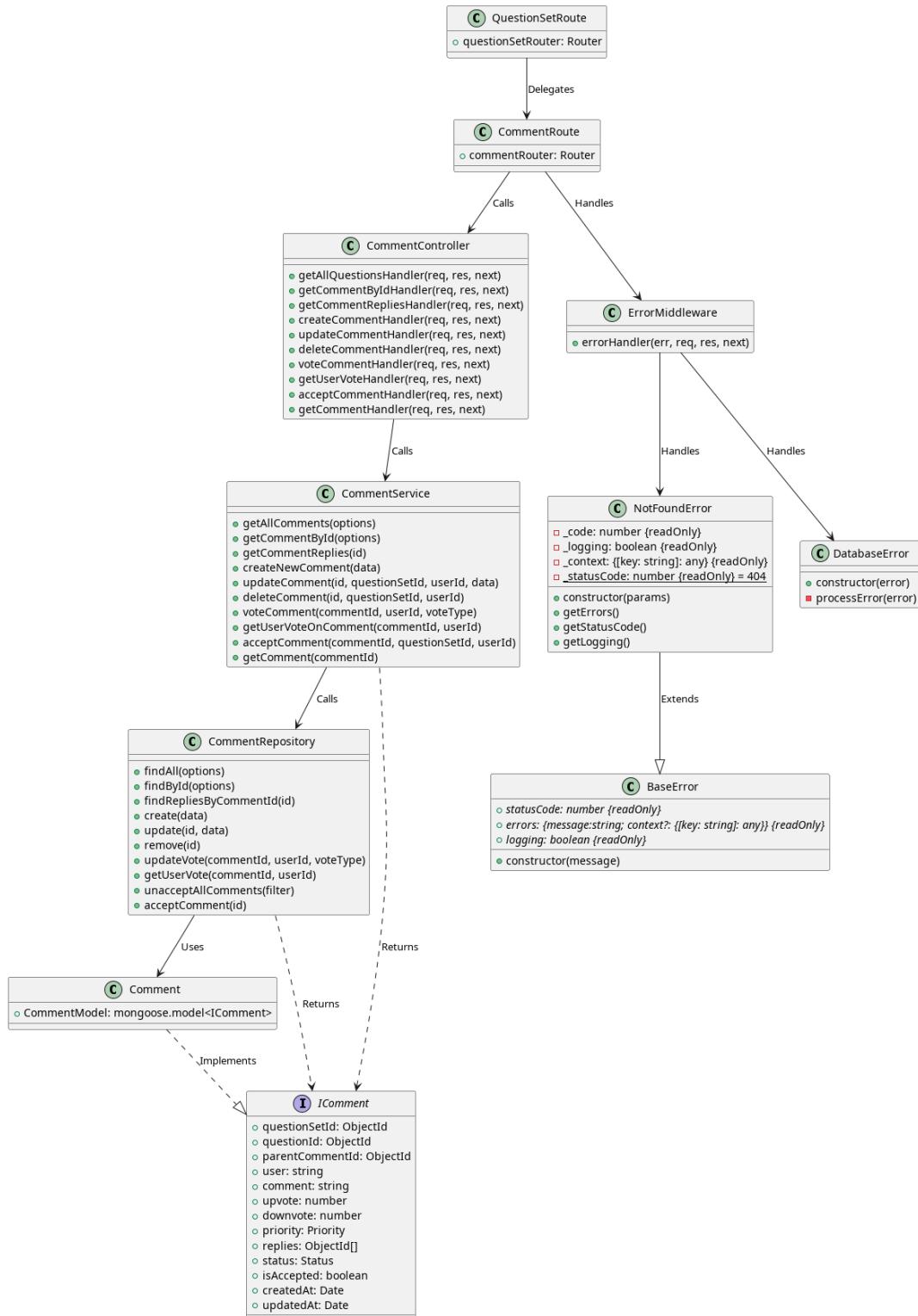


Figure 3.13.7.1: Filter comments²⁰⁷

²⁰⁷ [filter comments.png](#)

3.13.7.2 Sequence diagram

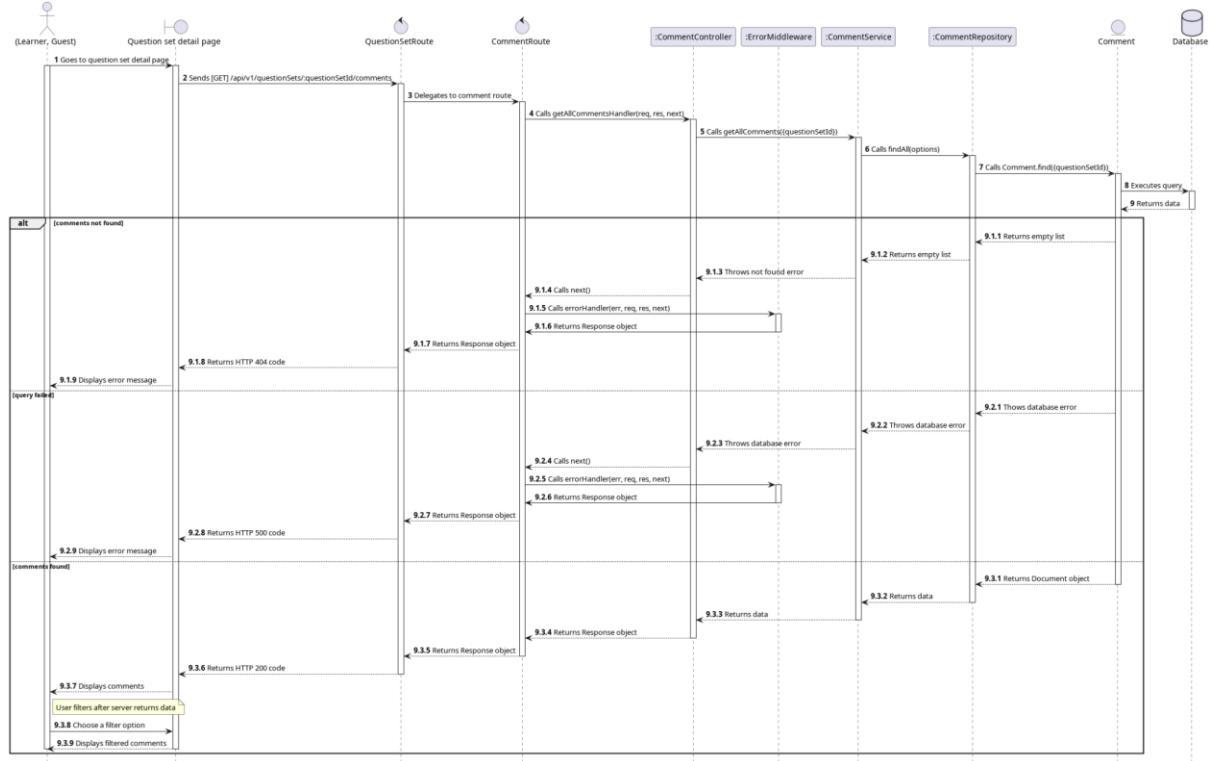


Figure 3.13.7.2: Filter comments²⁰⁸

208 [filter comments.png](#)

3.13.8 Mark comment

3.13.8.1 Class diagram

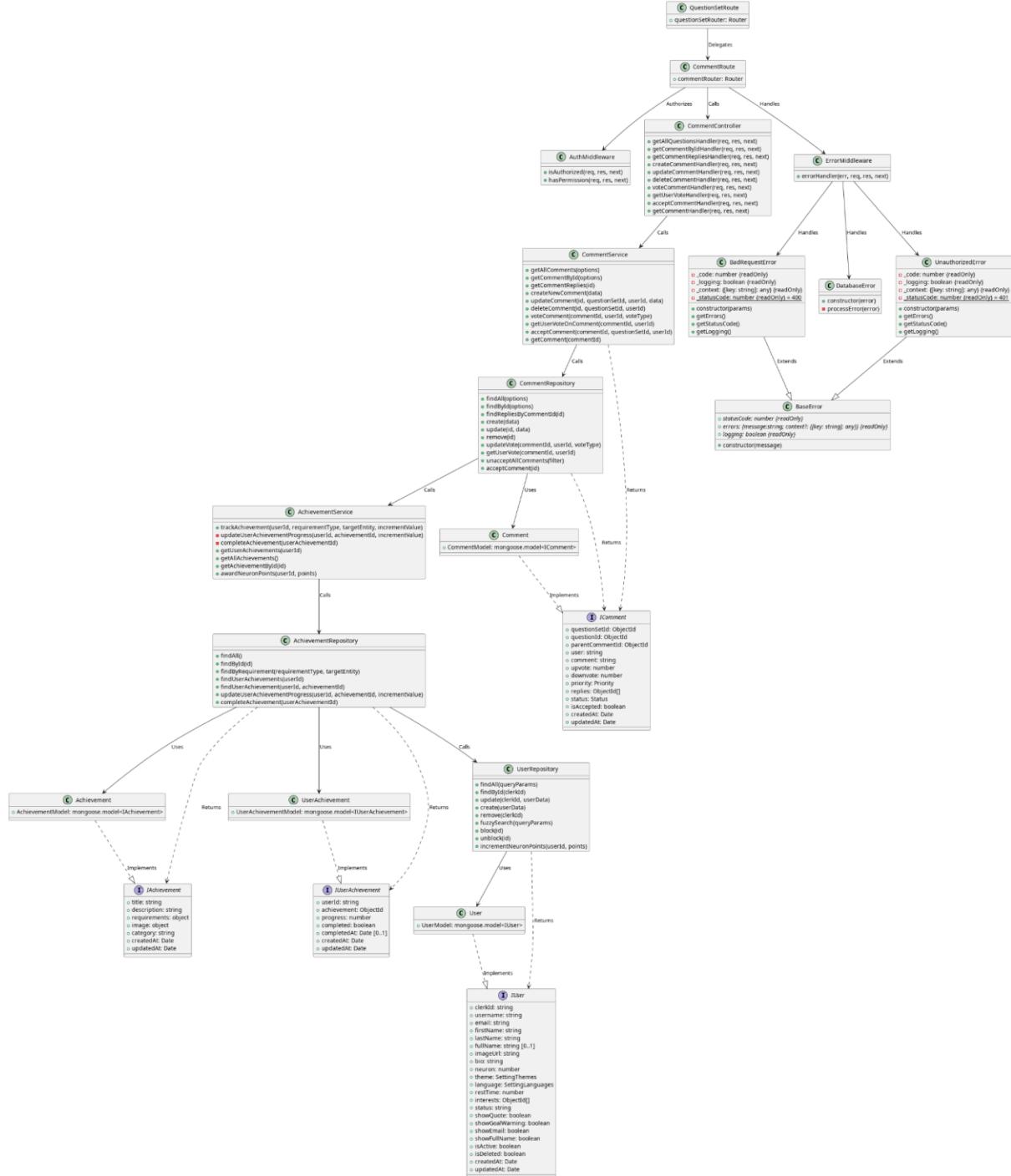


Figure 3.13.8.1: Mark comment²⁰⁹

209 mark comments.png

3.13.8.2 Sequence diagram

3.13.8.2.1 Mark comment

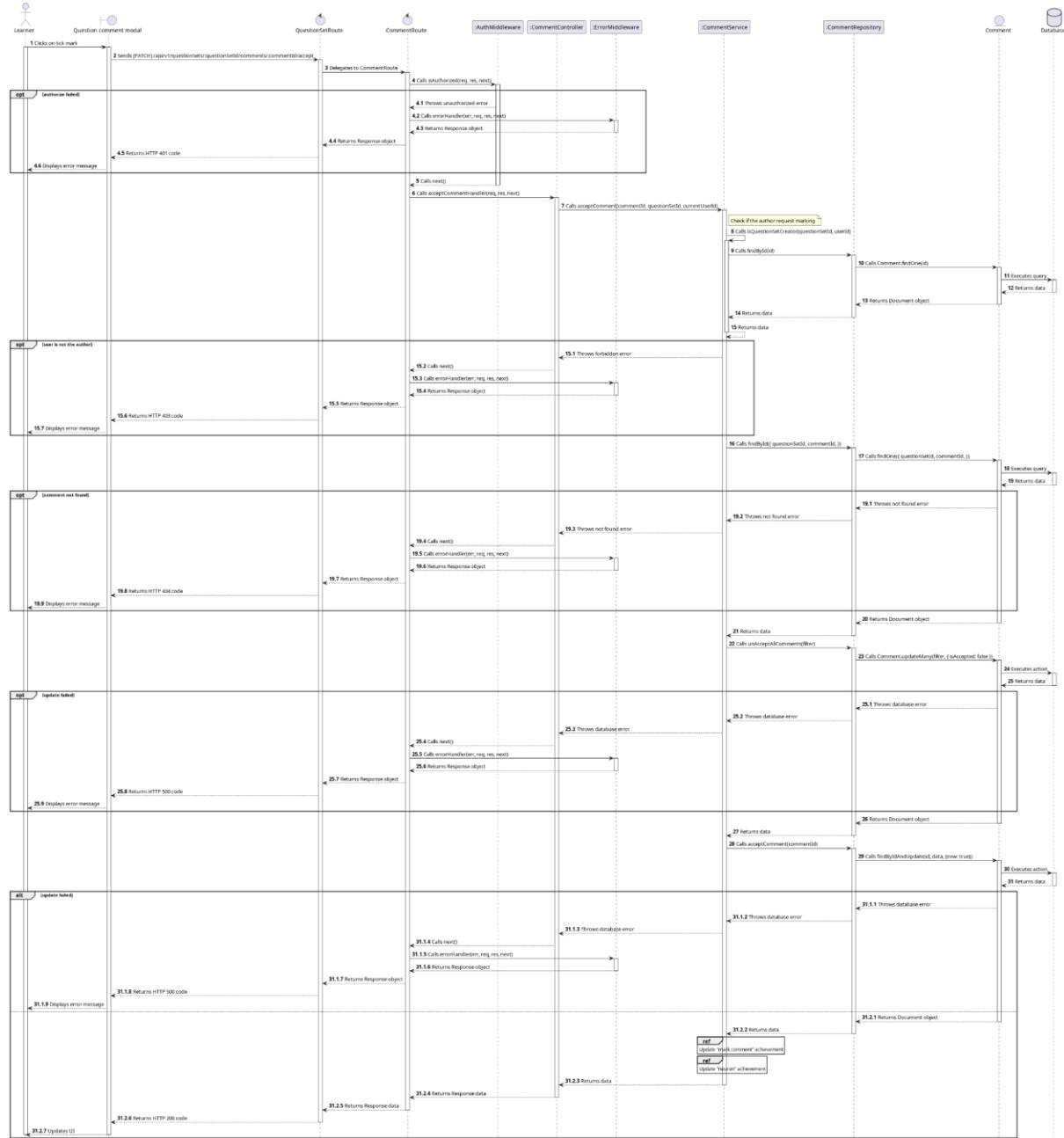


Figure 3.13.8.2.1: Mark comment²¹⁰

²¹⁰ mark comments.png

3.13.8.2.2 Update “mark comment” achievement

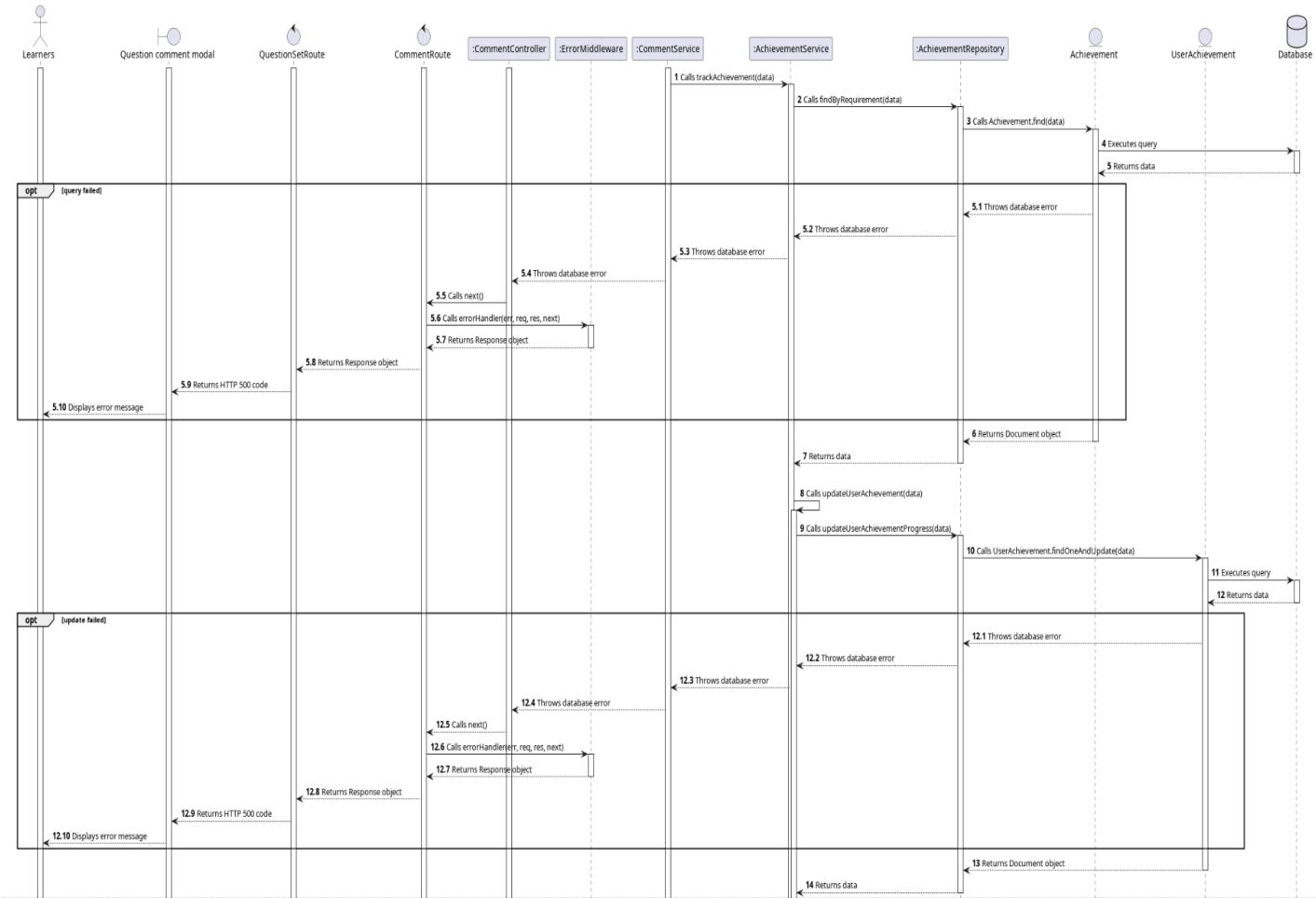


Figure 3.13.8.2.2.a: Update “mark comment” achievement - 1²¹¹

²¹¹ mark comment achievement.png

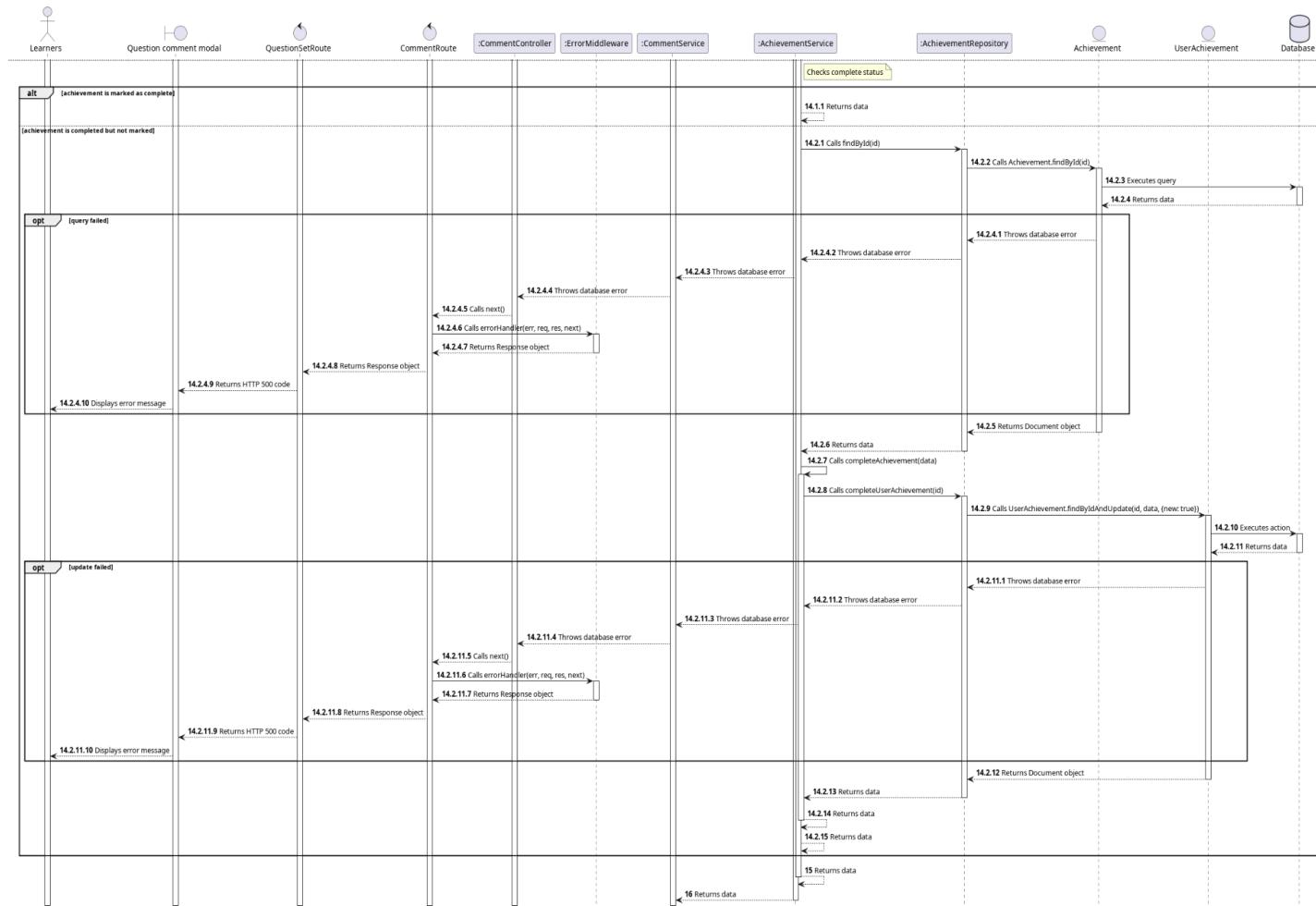


Figure 3.13.8.2.2.b: Update “mark comment” achievement - 2

3.13.8.2.3 Update “neuron” achievement

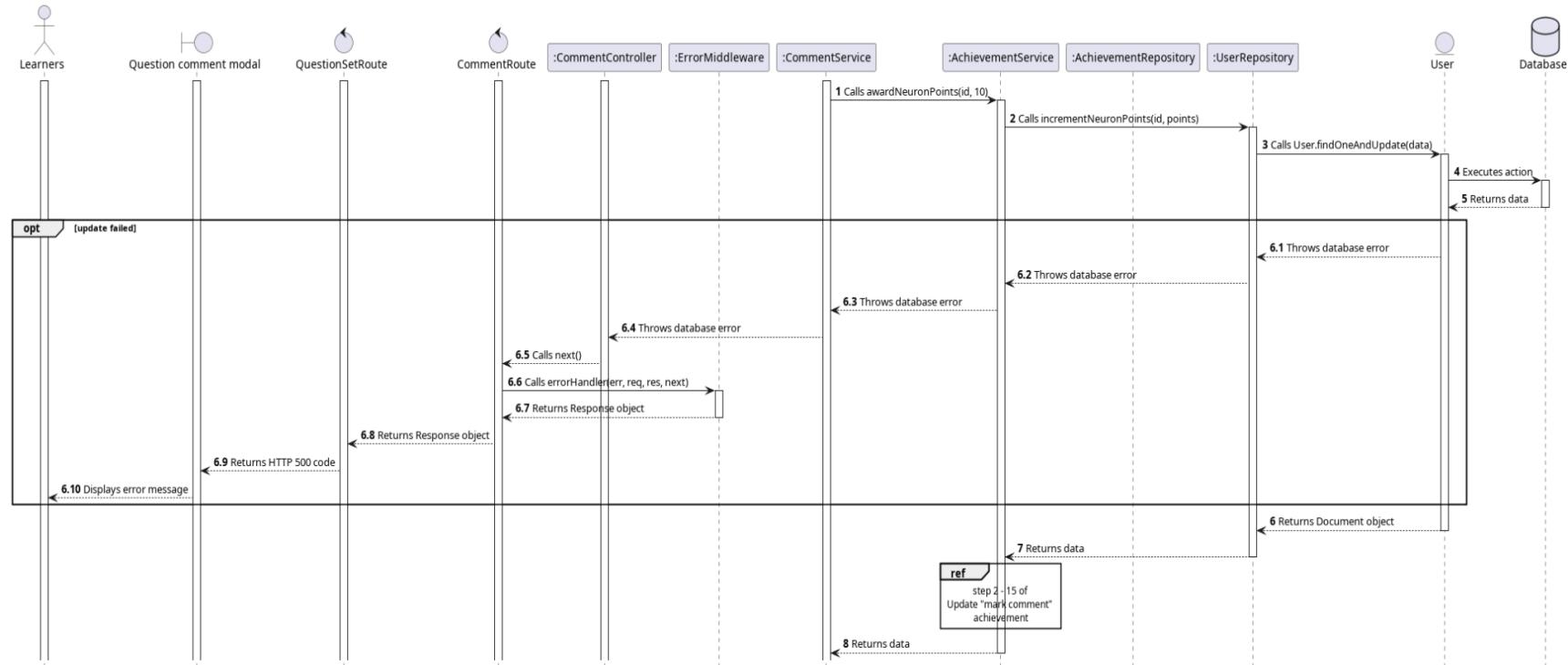


Figure 3.13.8.2.1: Update “neuron” achievement²¹²

²¹² [neuron achievement.png](#)

3.14 Learn Mode Interaction

3.14.1 Use learn mode

3.14.1.1 Class diagram

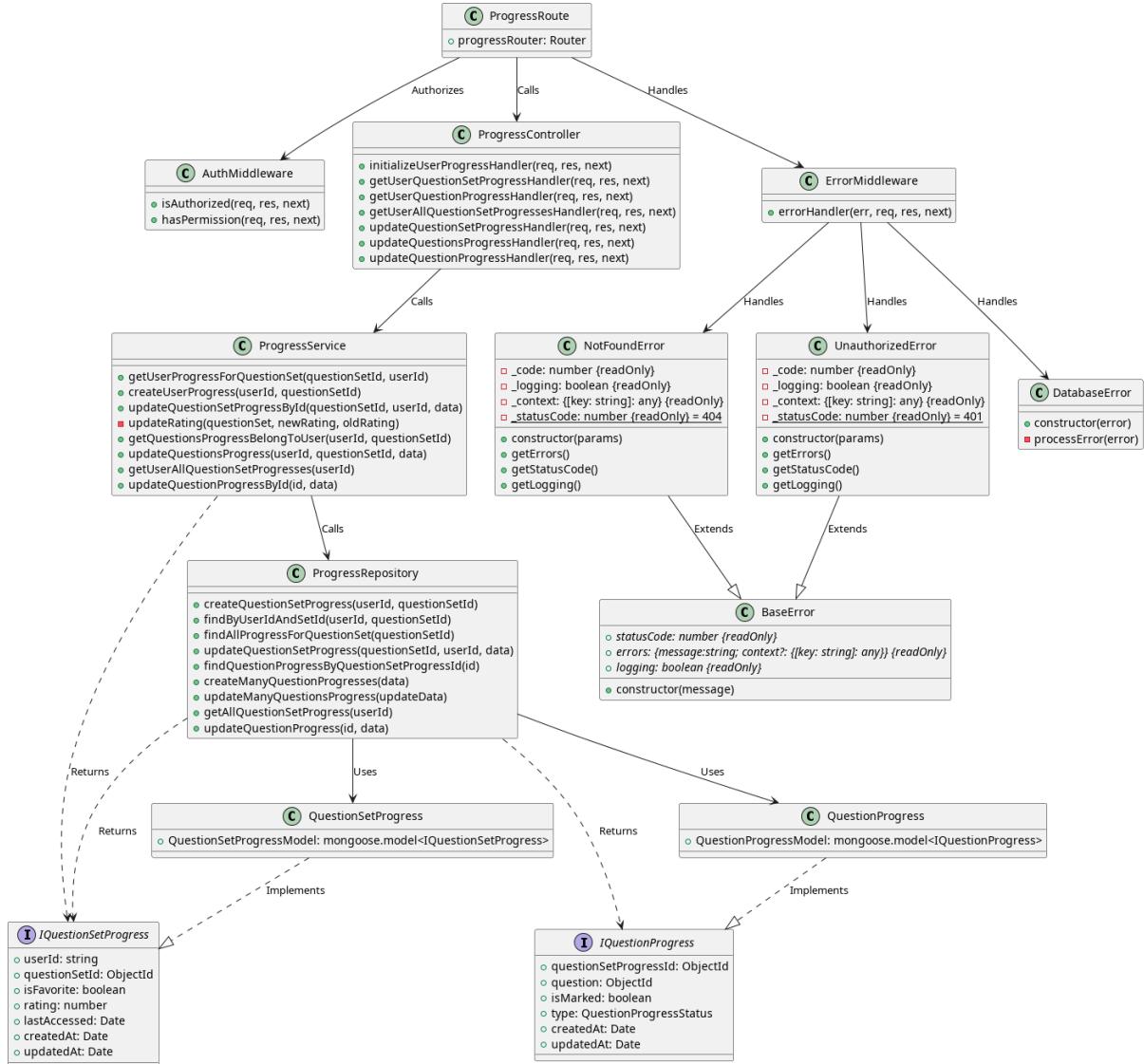


Figure 3.14.1.1: Use learn mode²¹³

²¹³ learn mode.png

3.14.1.2 Sequence diagram

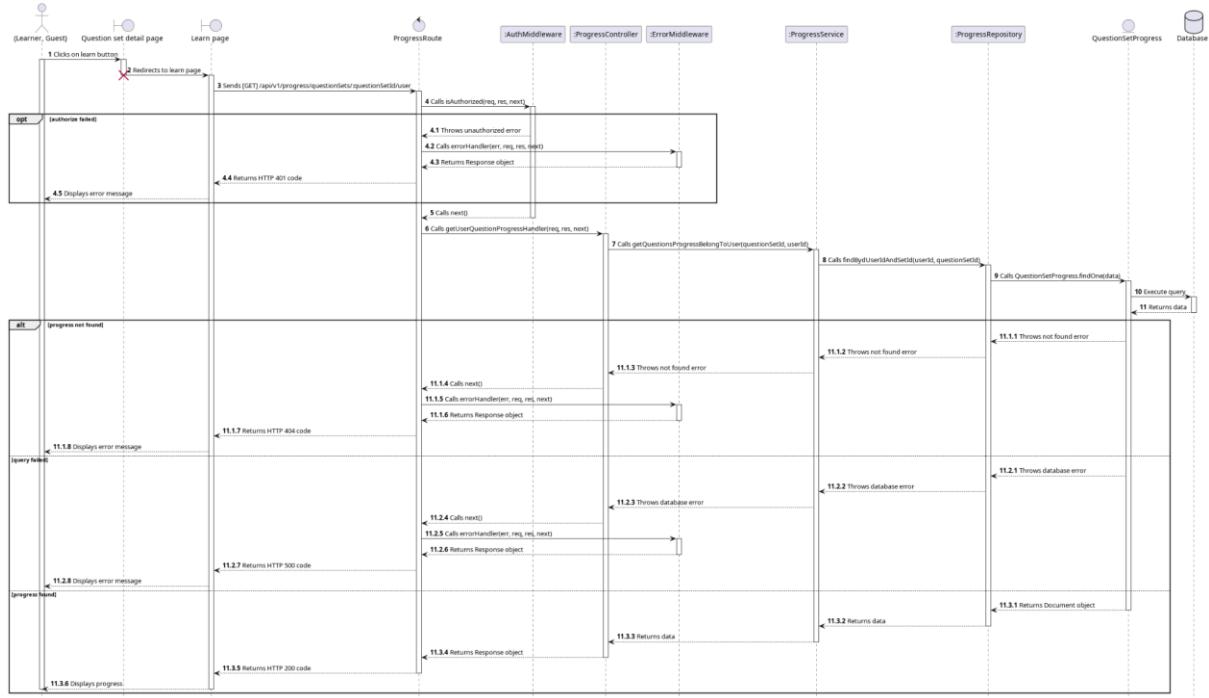


Figure 3.14.1.2: Use learn mode²¹⁴

²¹⁴ [learn mode.png](#)

3.14.2 Use practice mode

3.14.2.1 Class diagram

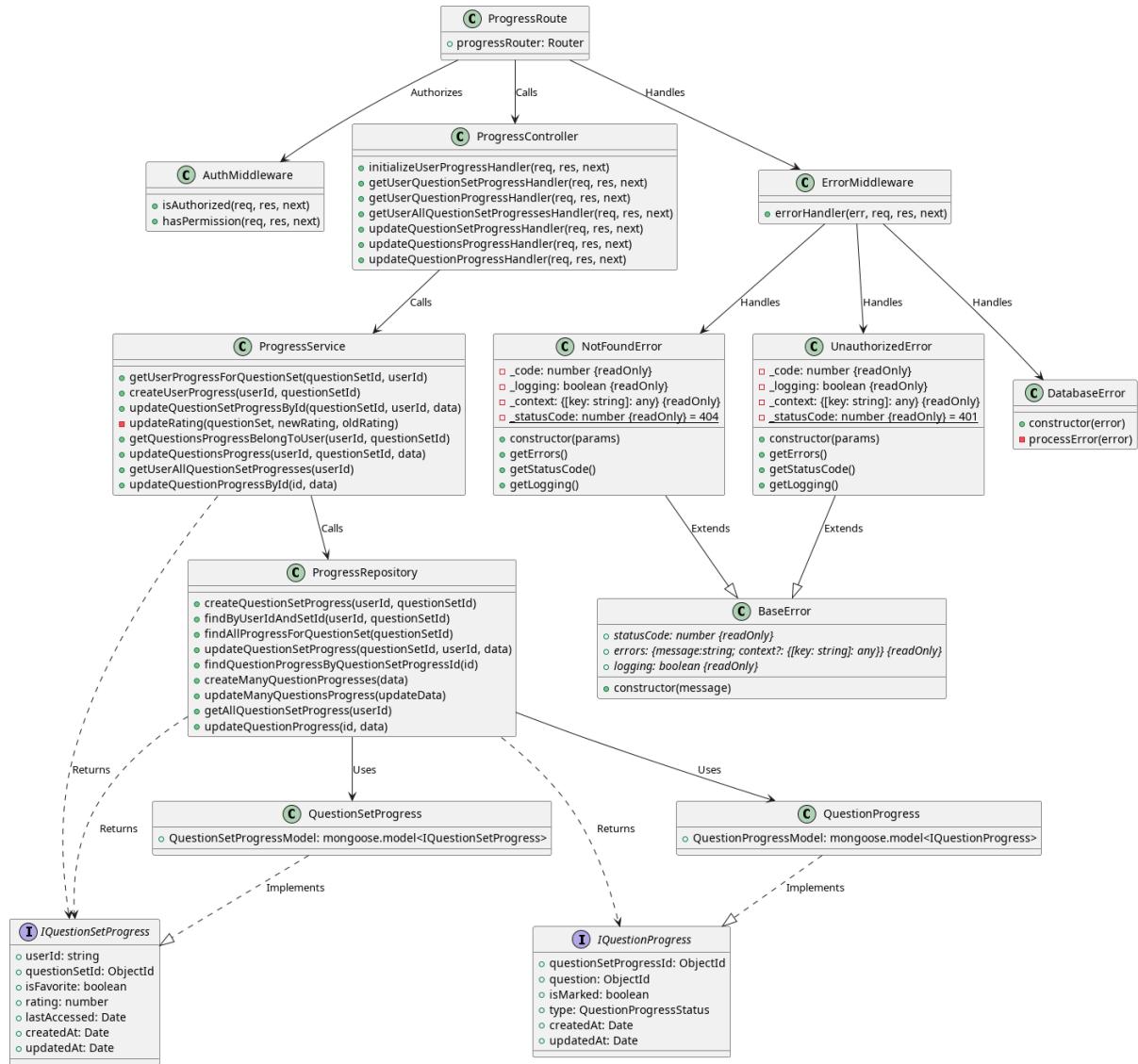


Figure 3.14.2.1: Use practice mode²¹⁵

215 learn mode.png

3.14.2.2 Sequence diagram

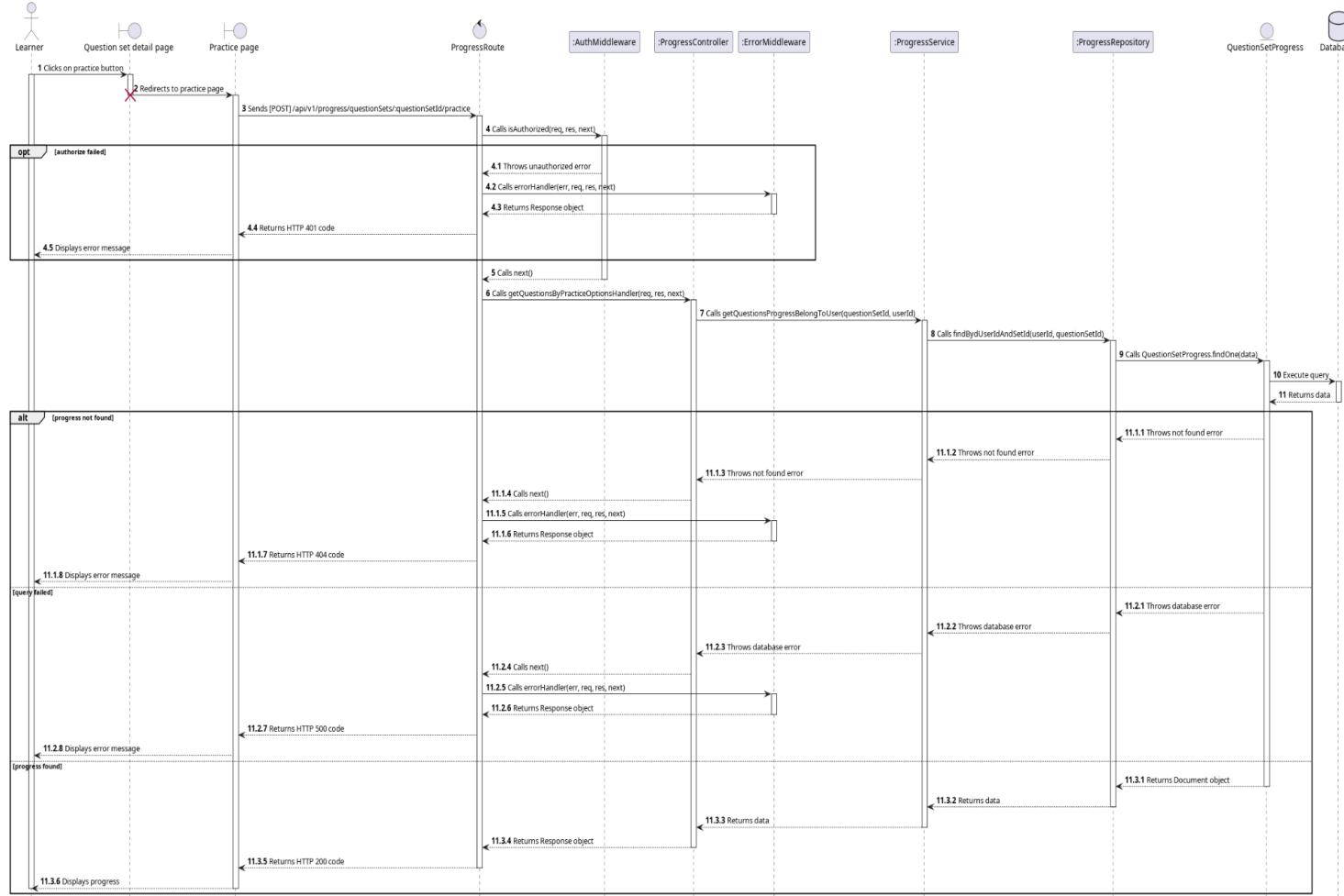


Figure 3.14.2.2.a: Get practice question - 1²¹⁶

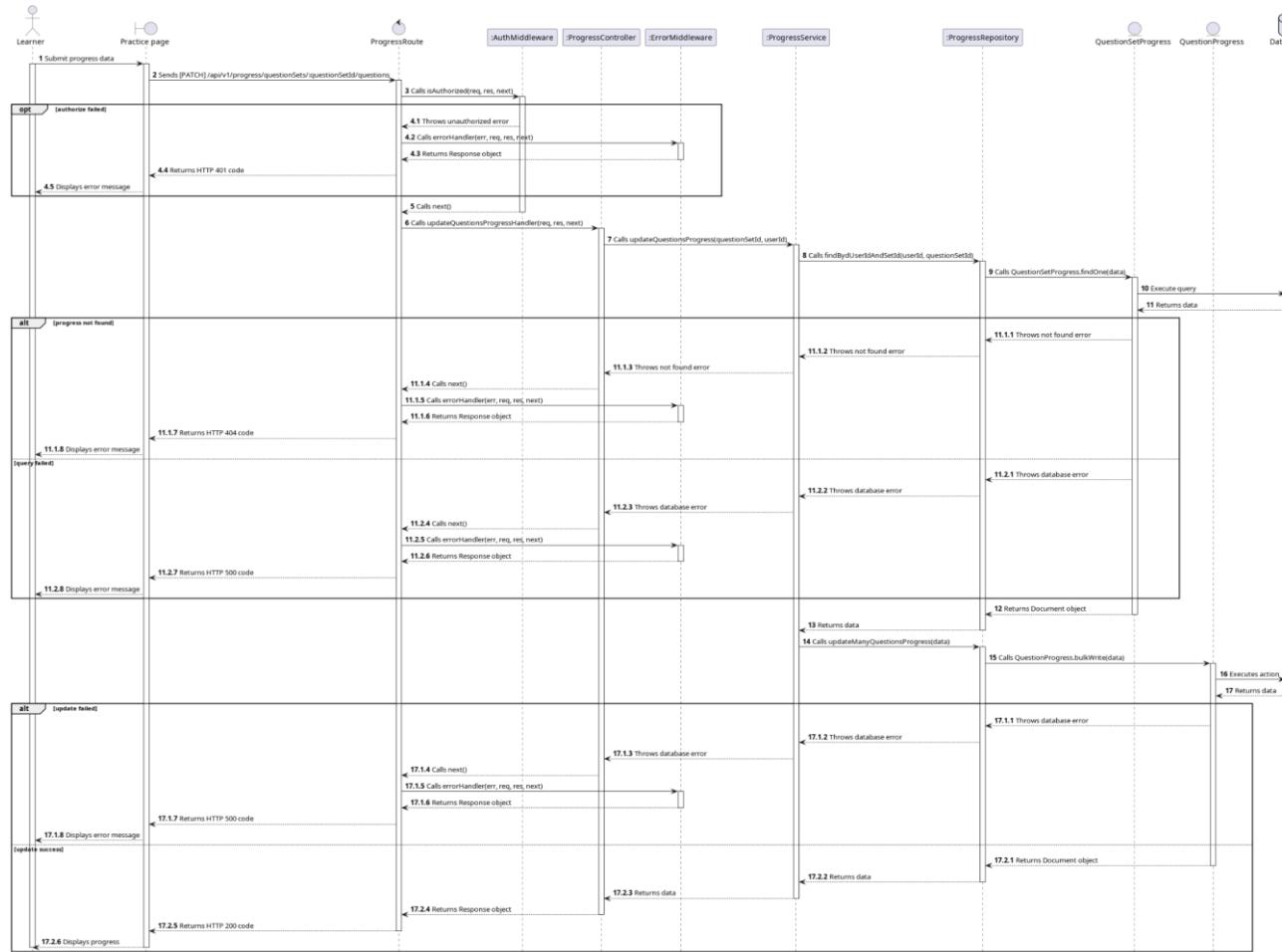


Figure 3.14.2.2.b: Update question progress - 2²¹⁷

²¹⁷ [update practice.png](#)

3.14.3 Use matching mode

3.14.3.1 Class diagram



Figure 3.14.3.1: Use matching mode²¹⁸

²¹⁸ use matching mode.png

3.14.3.2 Sequence diagram

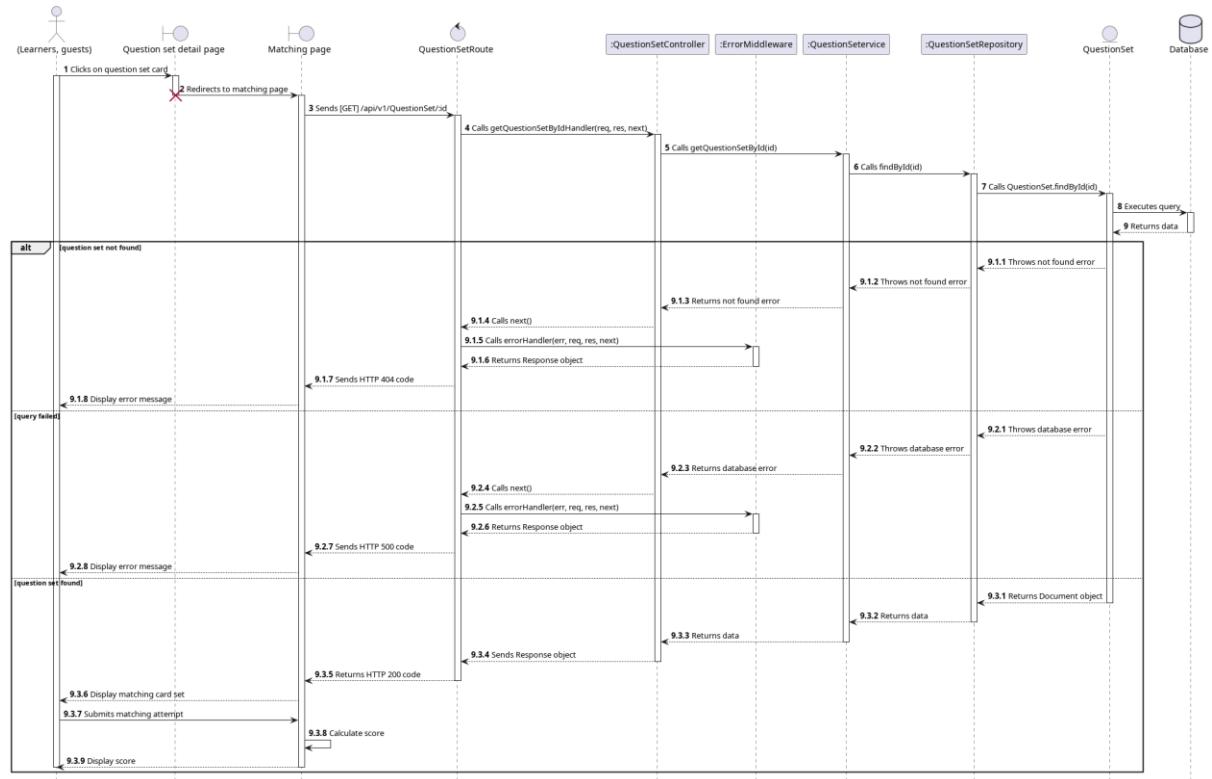


Figure 3.14.2.2: Use matching mode²¹⁹

219 use matching mode.png

3.15 Quiz Mode Interaction

3.15.1 Use quiz mode

3.15.1.1 Class diagram

3.15.1.1.1 Create quiz

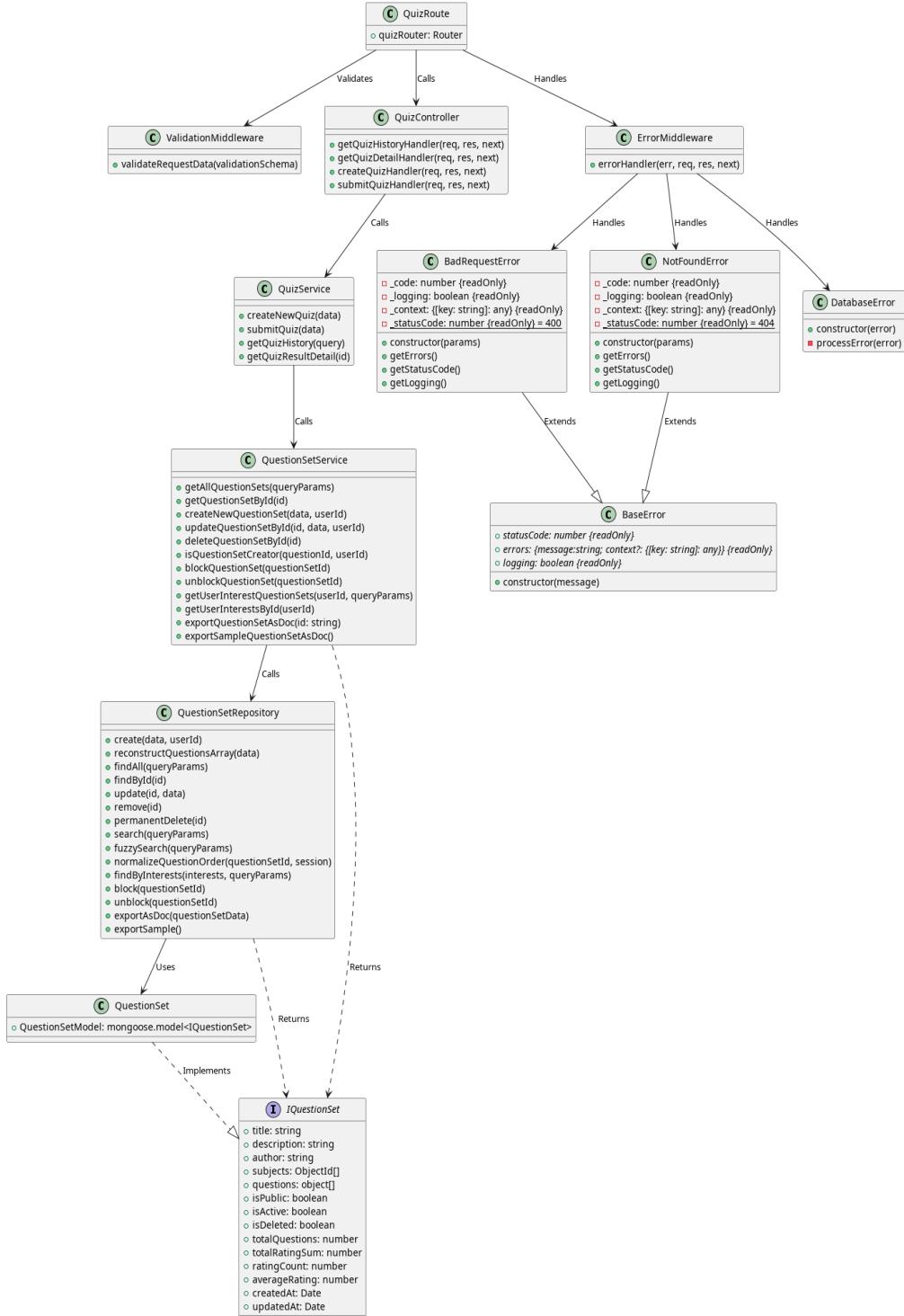


Figure 3.15.1.1.1: Create quiz²²⁰

²²⁰ [create quiz.png](#)

3.15.1.1.2 Submit quiz

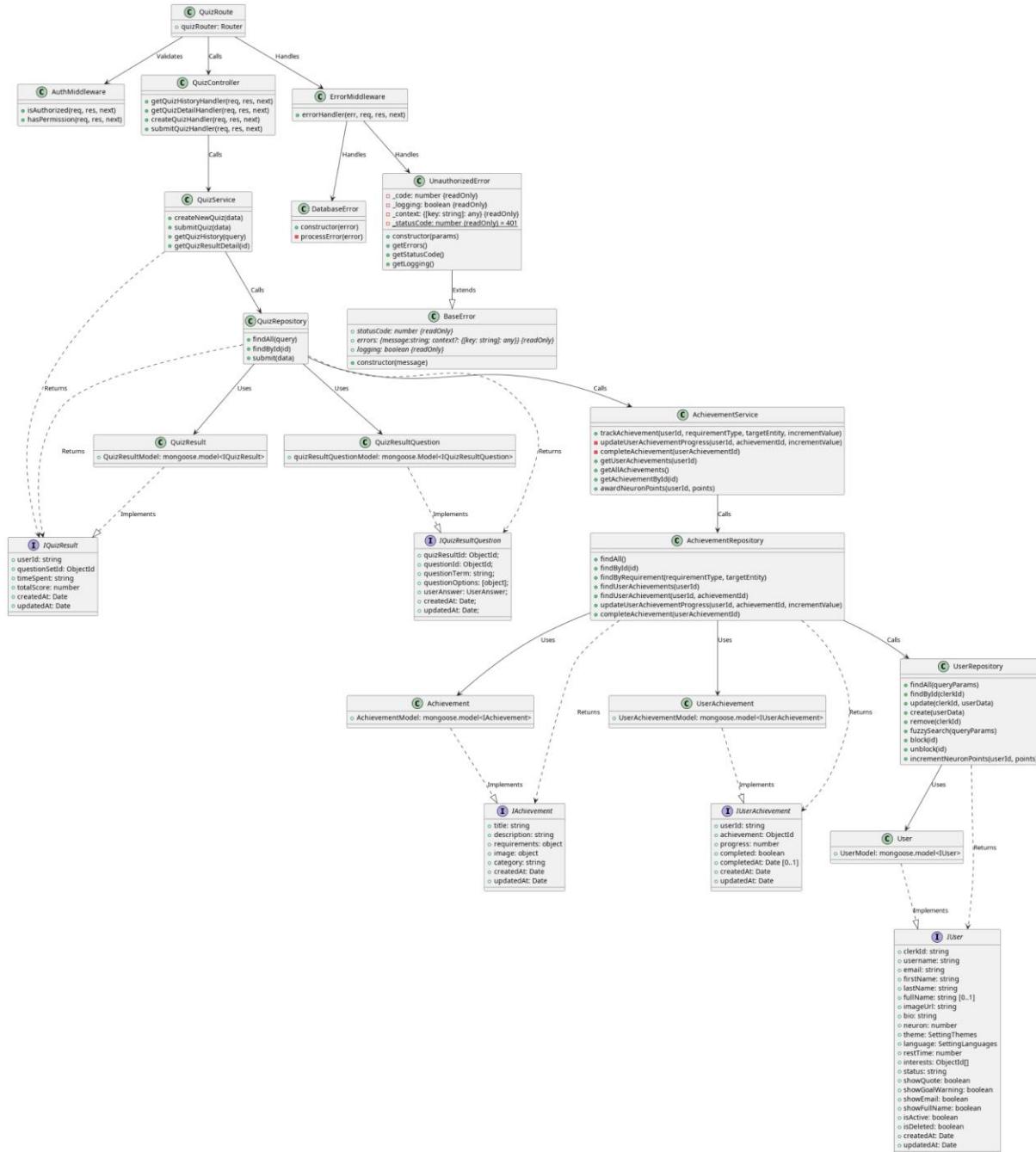


Figure 3.15.1.1.2.a: Learner submits quiz²²¹

²²¹ submit quiz.png

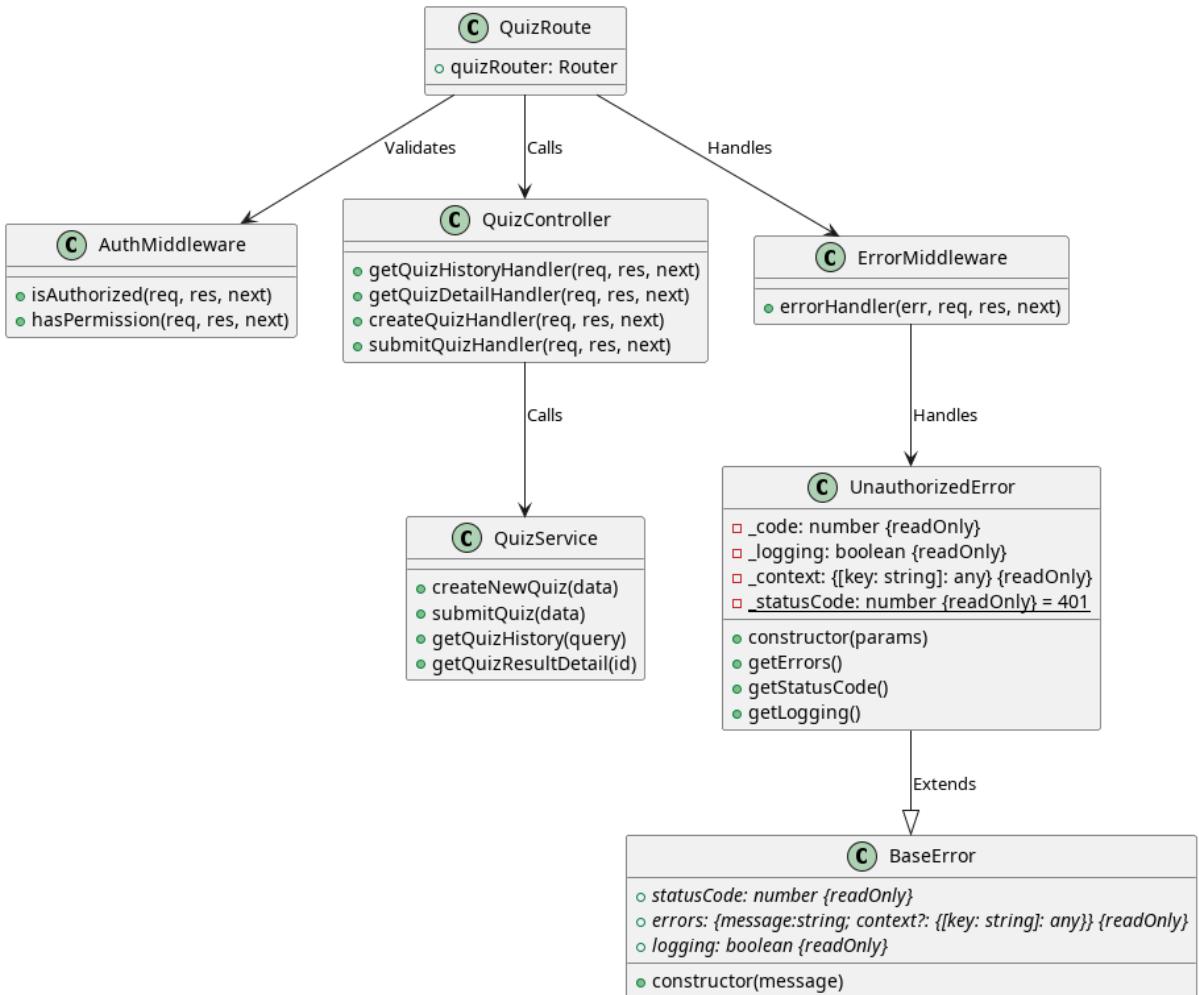


Figure 3.15.1.1.2.b: Guest submits quiz²²²

²²² submit quiz.png

3.15.1.2 Sequence diagram

3.15.1.2.1 Create quiz

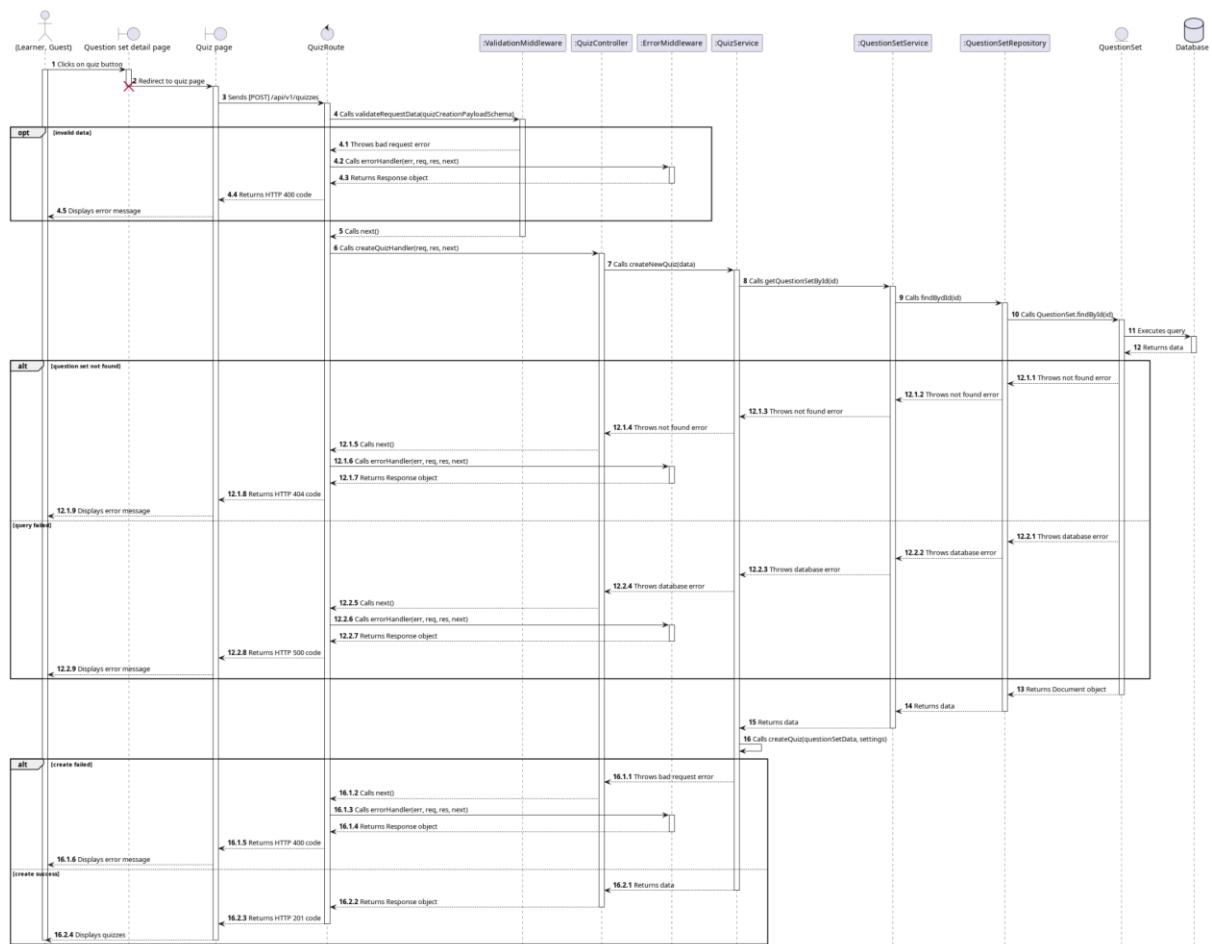


Figure 3.15.1.2.1: Create quiz²²³

223 create quiz.png

3.15.1.2.2 Submit quiz

3.15.1.2.2.1 Submit quiz

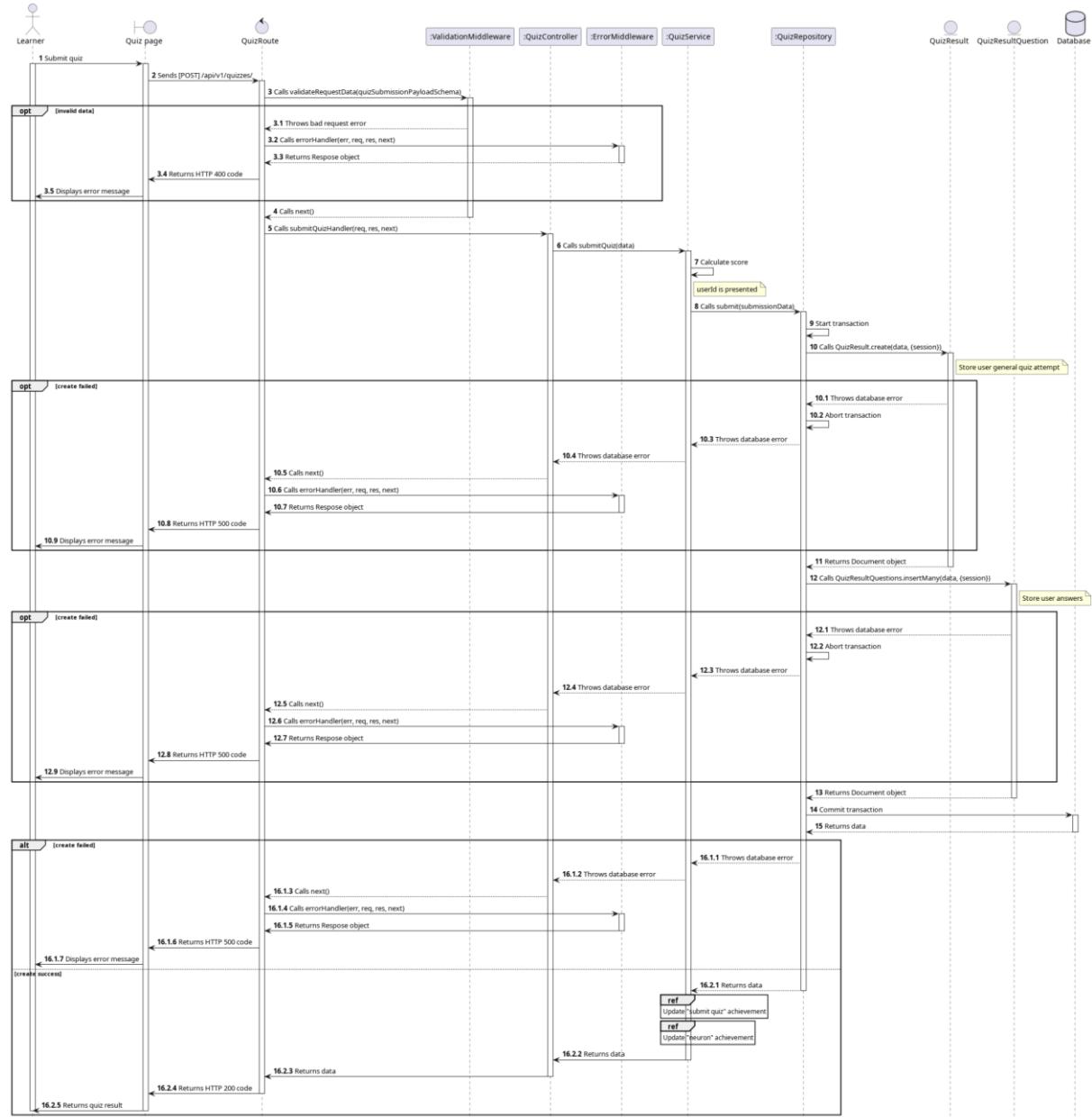


Figure 3.15.1.2.2.1.a: Learner submit quiz²²⁴

²²⁴ submit quiz.png

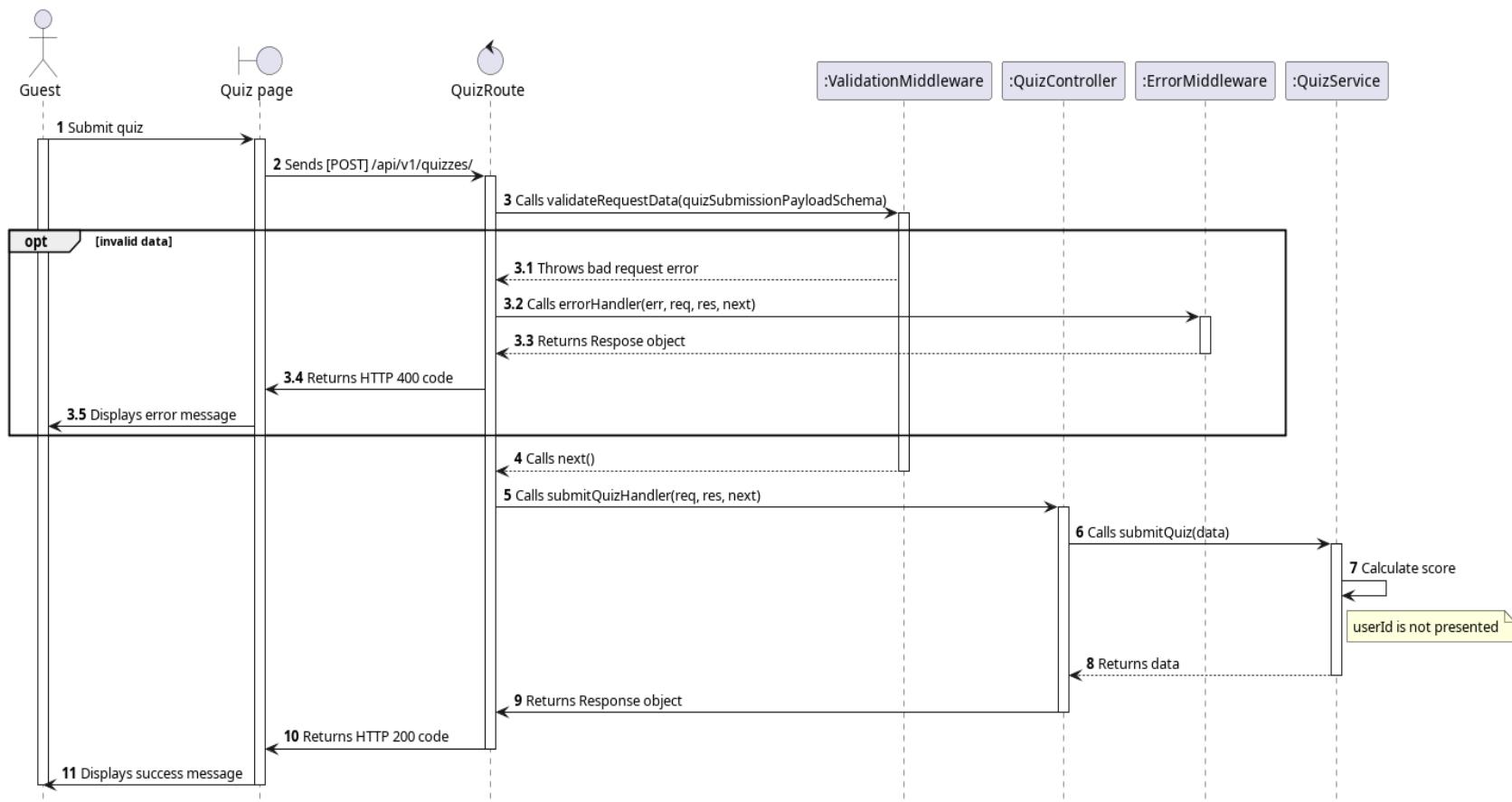


Figure 3.15.1.2.2.b: Guest submit quiz

3.15.1.2.2.2 Update “submit quiz” achievement

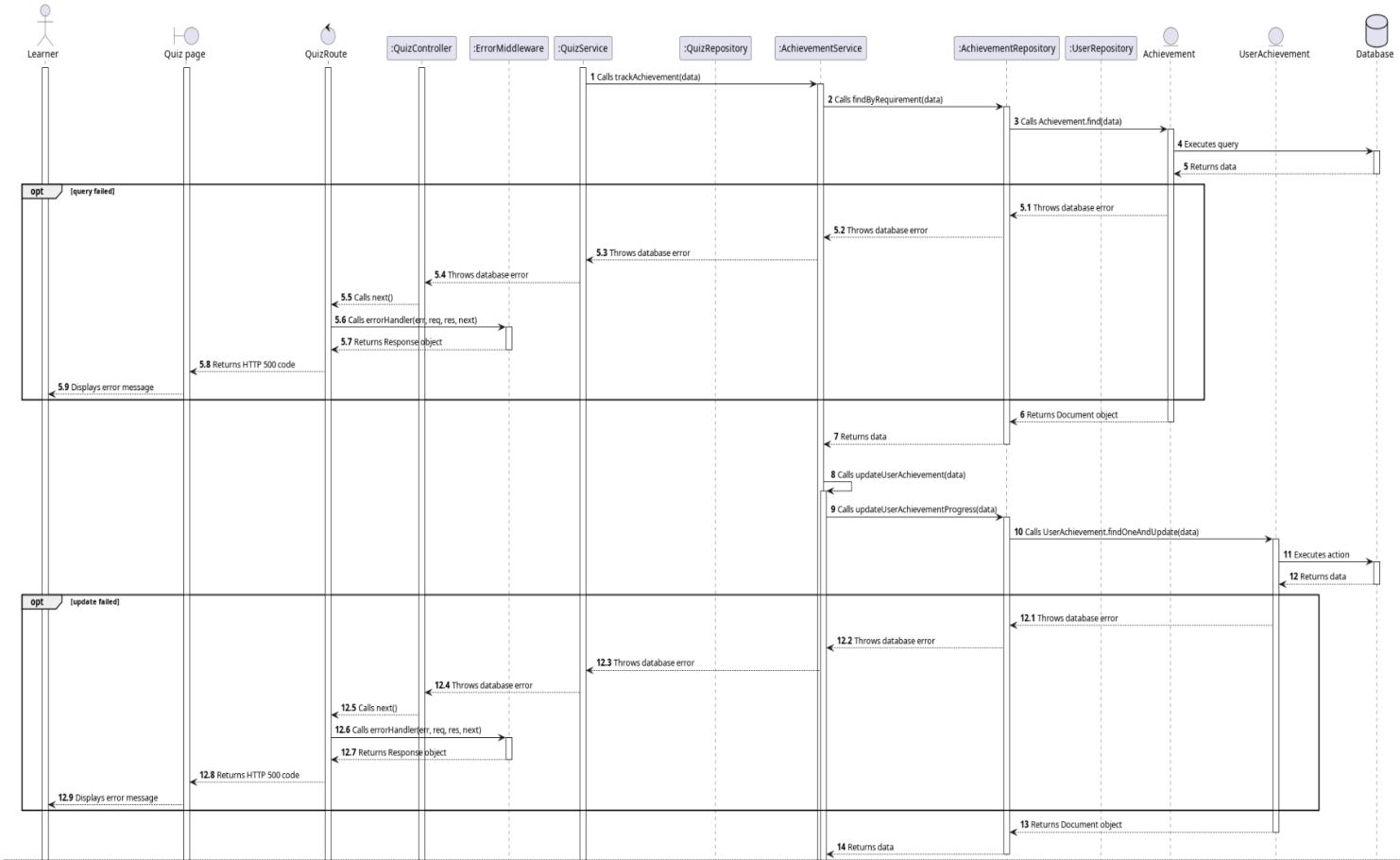


Figure 3.15.1.2.2.1.2.a: Update “submit quiz” achievement - 1²²⁵

225 submit quiz.png

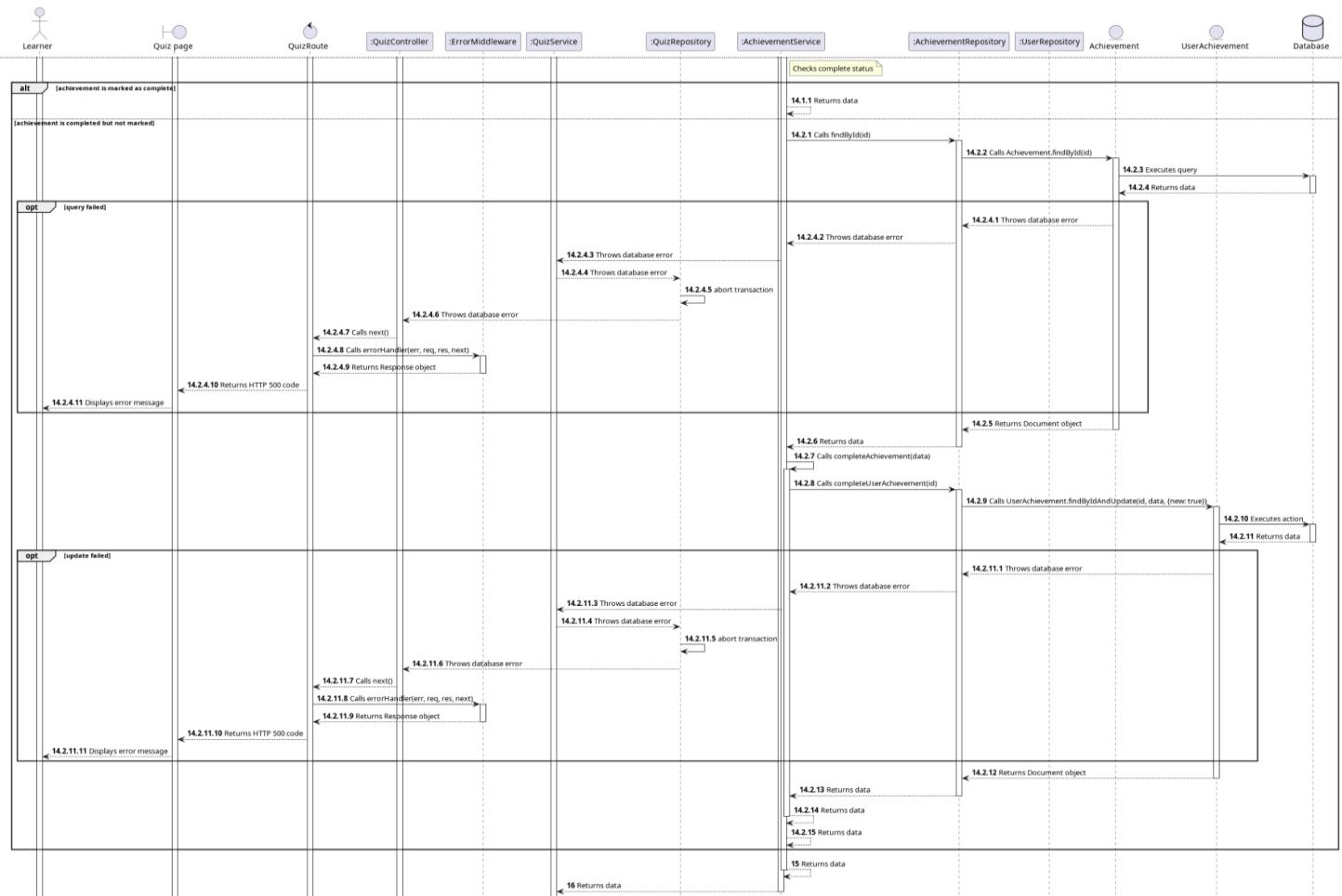


Figure 3.15.1.2.2.1.2.b: Update “submit quiz” achievement - 2

3.15.1.2.2.3 Update “neuron” achievement

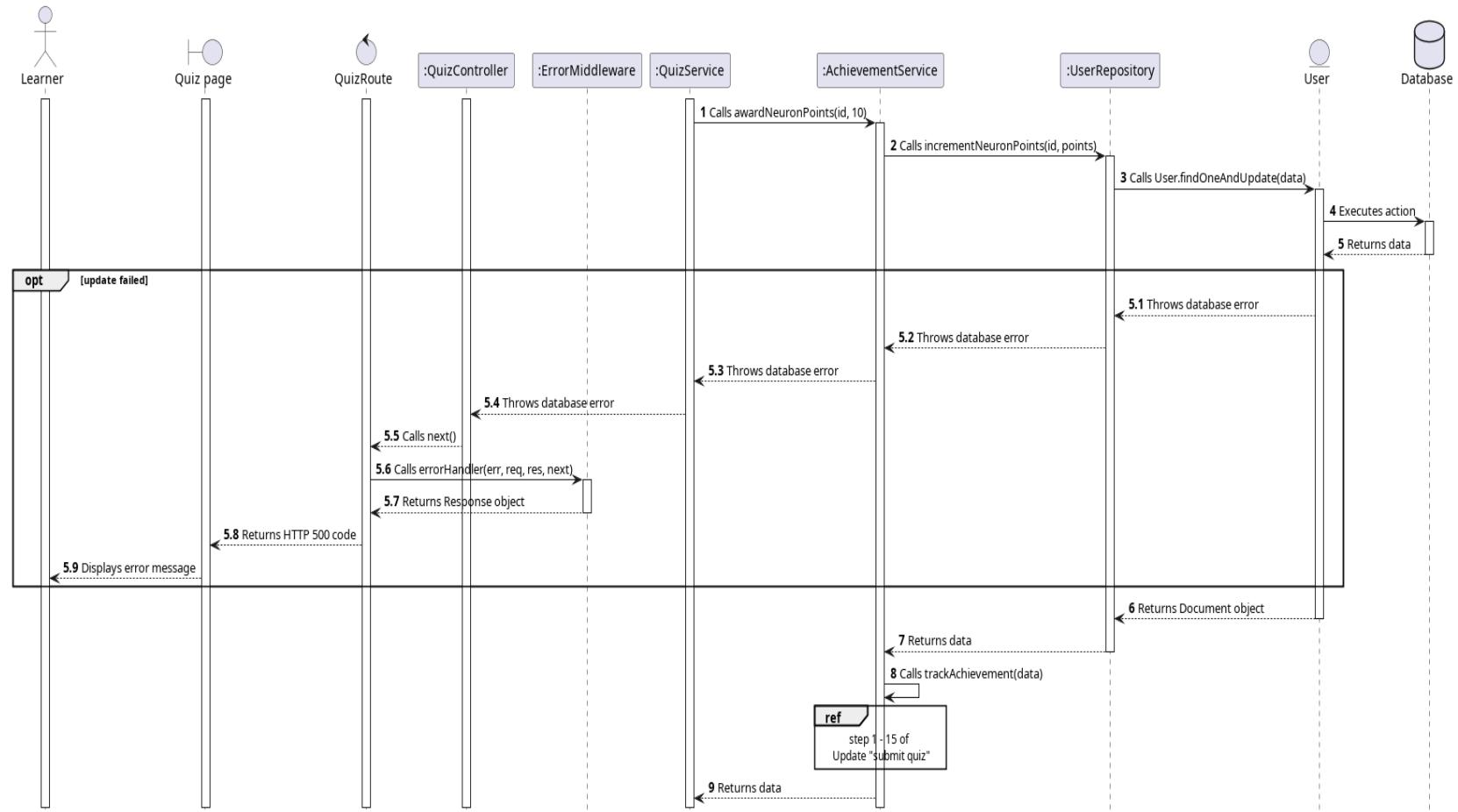


Figure 3.15.1.2.2.1.3: Update “neuron” achievement²²⁶

²²⁶ [neuron achievement.png](#)

3.15.2 Revisit quiz attempts

3.15.2.1 Class diagram

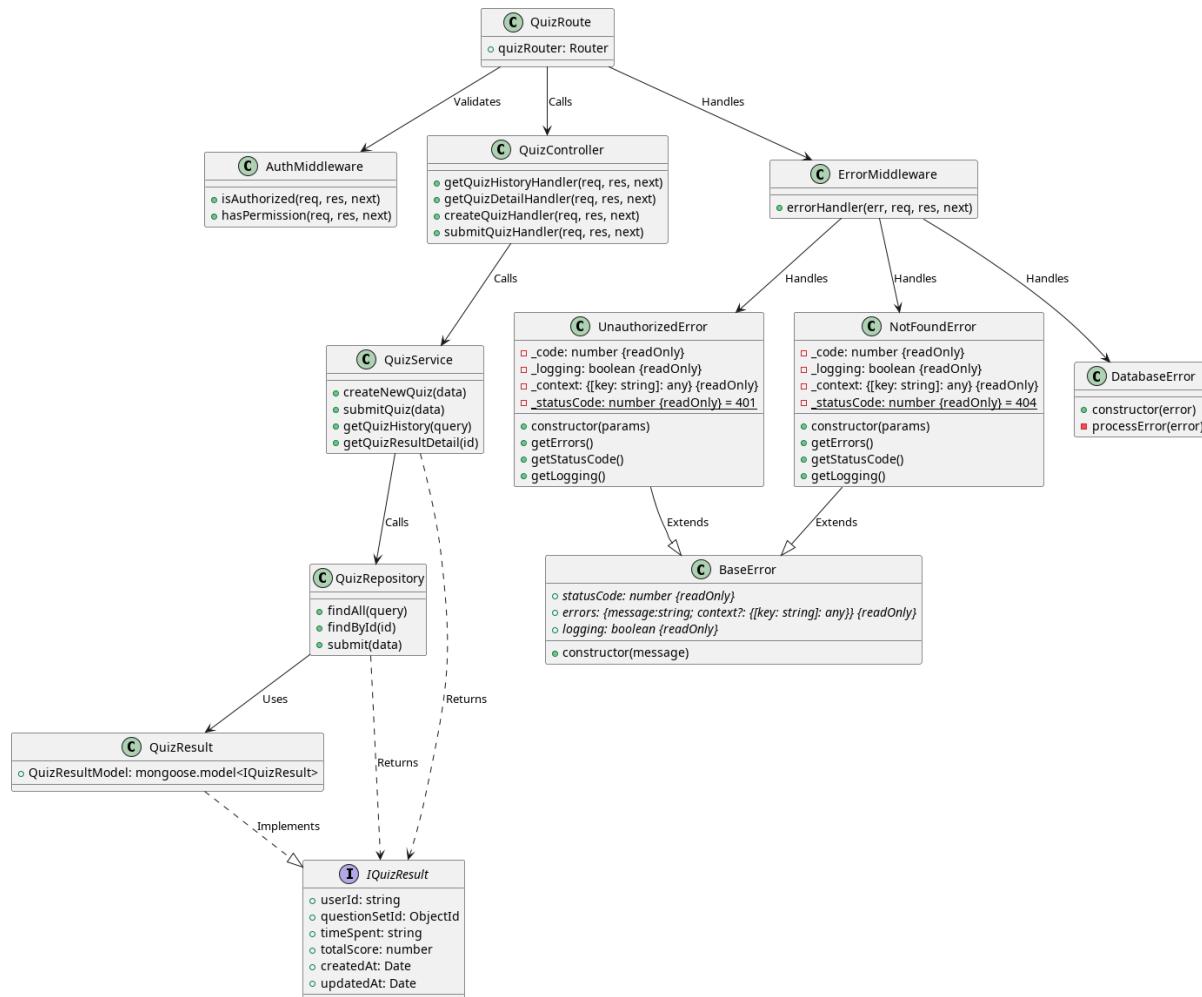


Figure 3.15.2.1: Revisit quiz attempts²²⁷

²²⁷ [revisit.png](#)

3.15.2.2 Sequence diagram

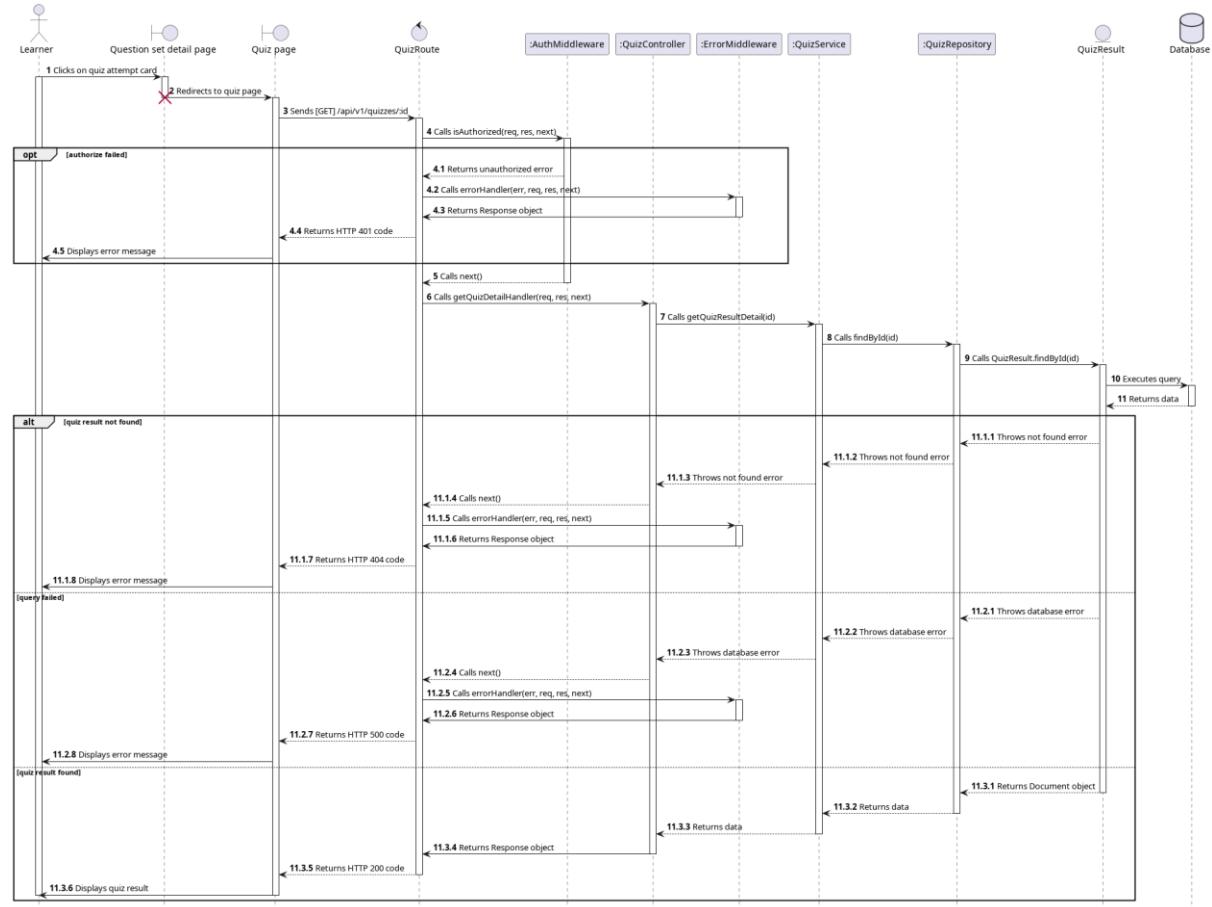


Figure 3.15.2: Revisit quiz attempts²²⁸

228 [revisit quiz attempt.png](#)

3.15.3 View quiz statistics

3.15.3.1 Class diagram

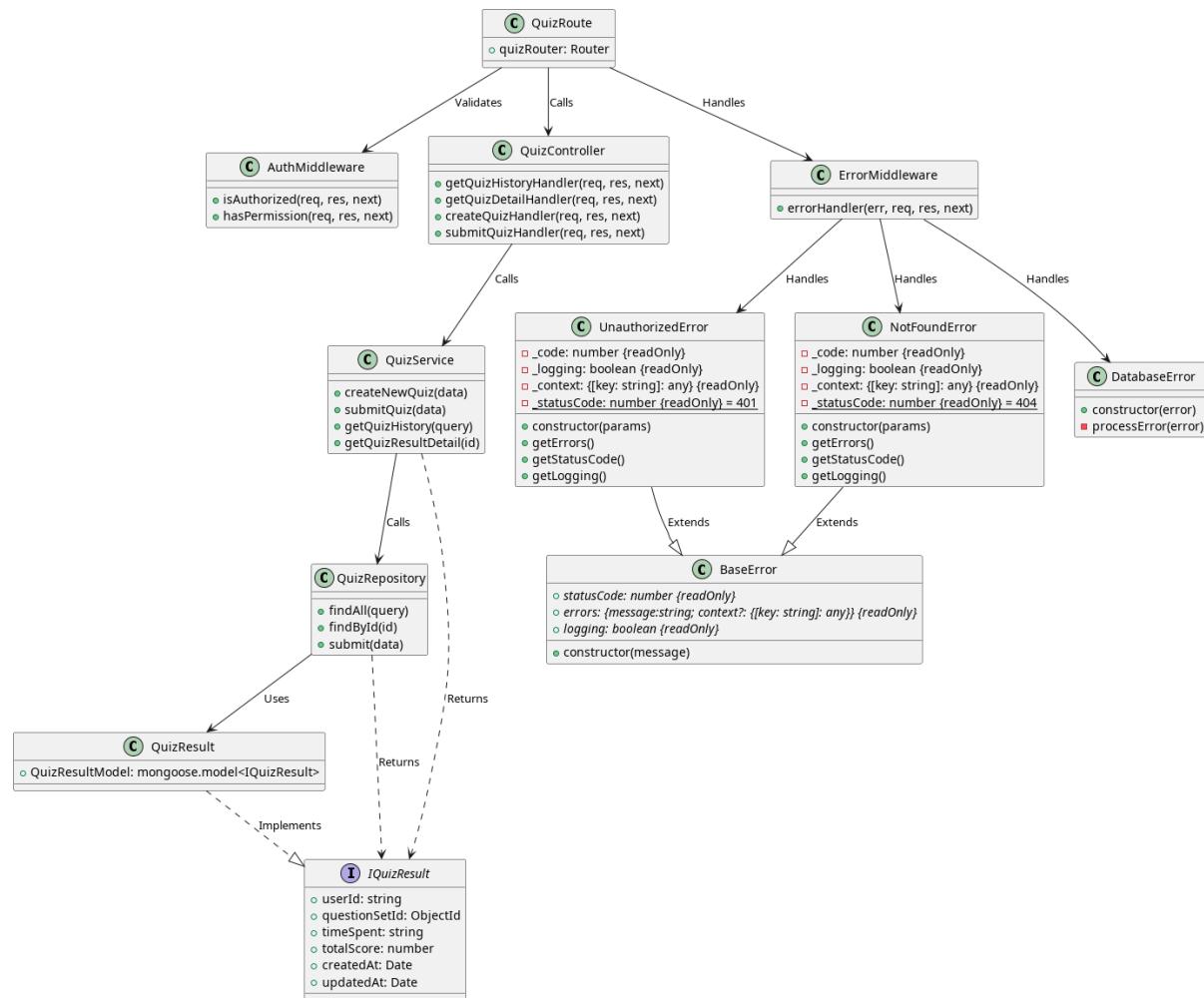


Figure 3.15.3.1: View quiz statistics²²⁹

²²⁹ [view statistics.png](#)

3.15.3.2 Sequence diagram

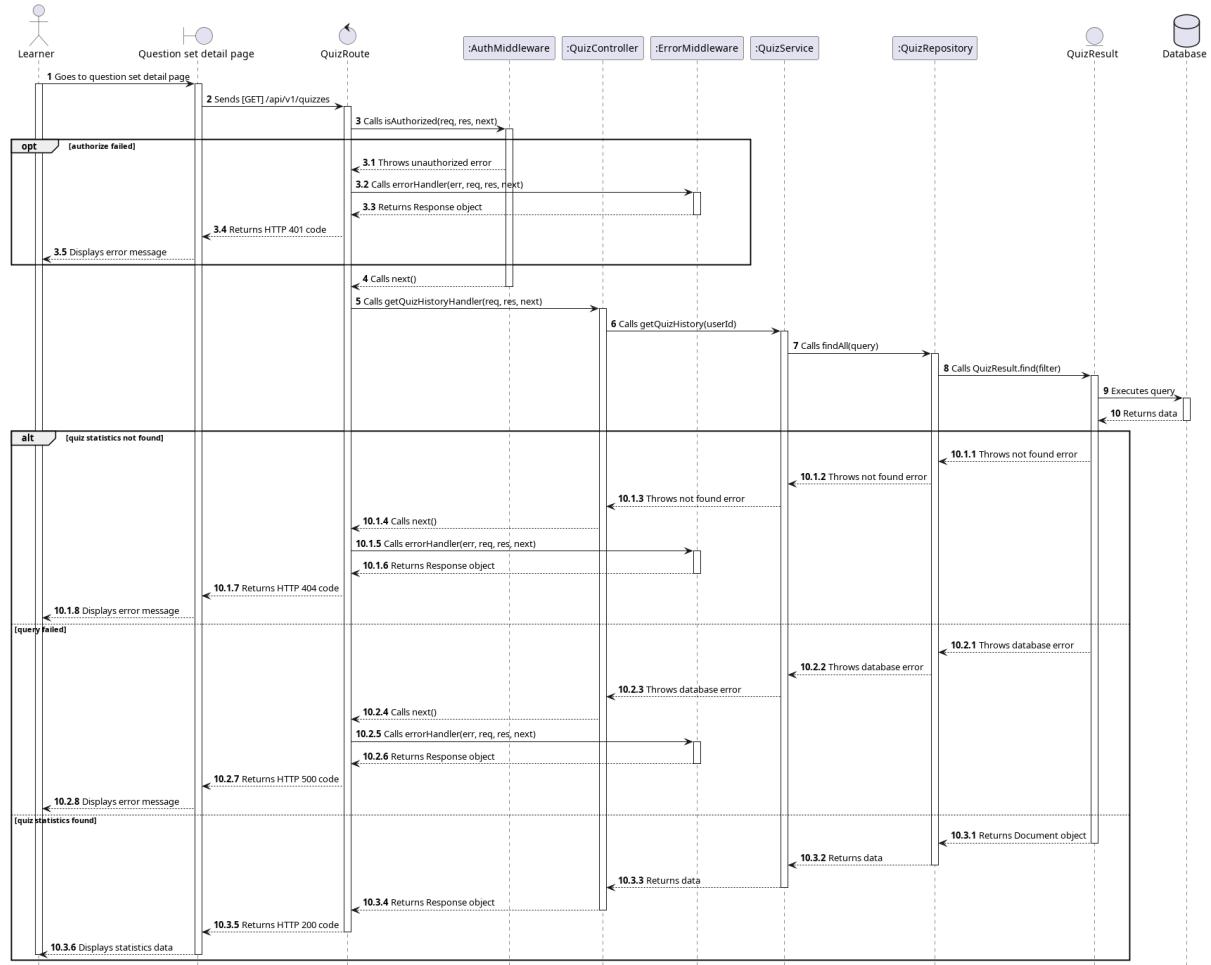


Figure 3.15.3.2: View quiz statistics²³⁰

230 [quiz statistics.png](#)

3.16 Statistics Interaction

3.16.1 View statistics

3.16.1.1 Class diagram

3.16.1.1.1 Get total users

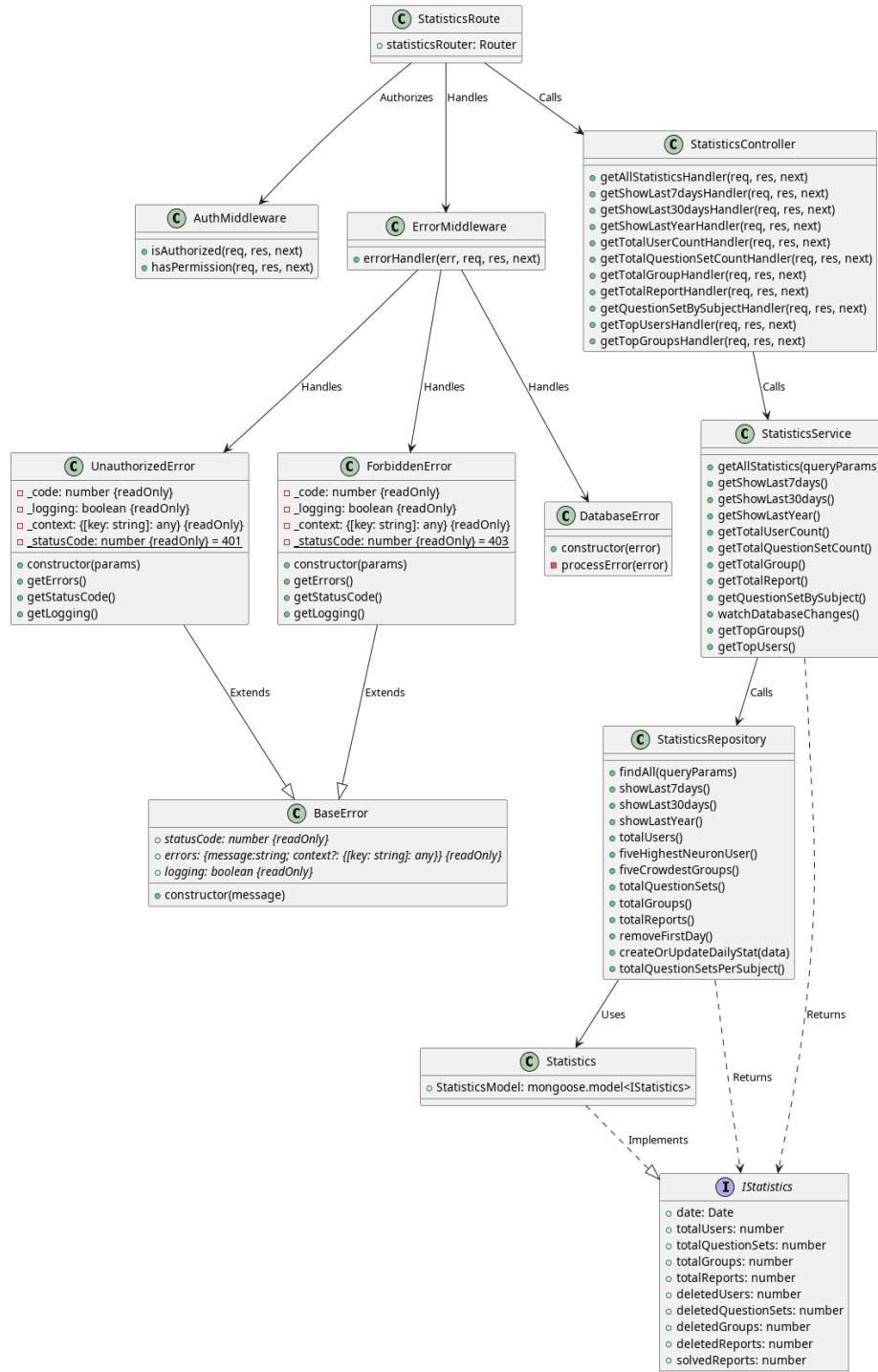


Figure 3.16.1.1.1: Get total users²³¹

²³¹ [statistics.png](#)

3.16.1.1.2 Get total question sets

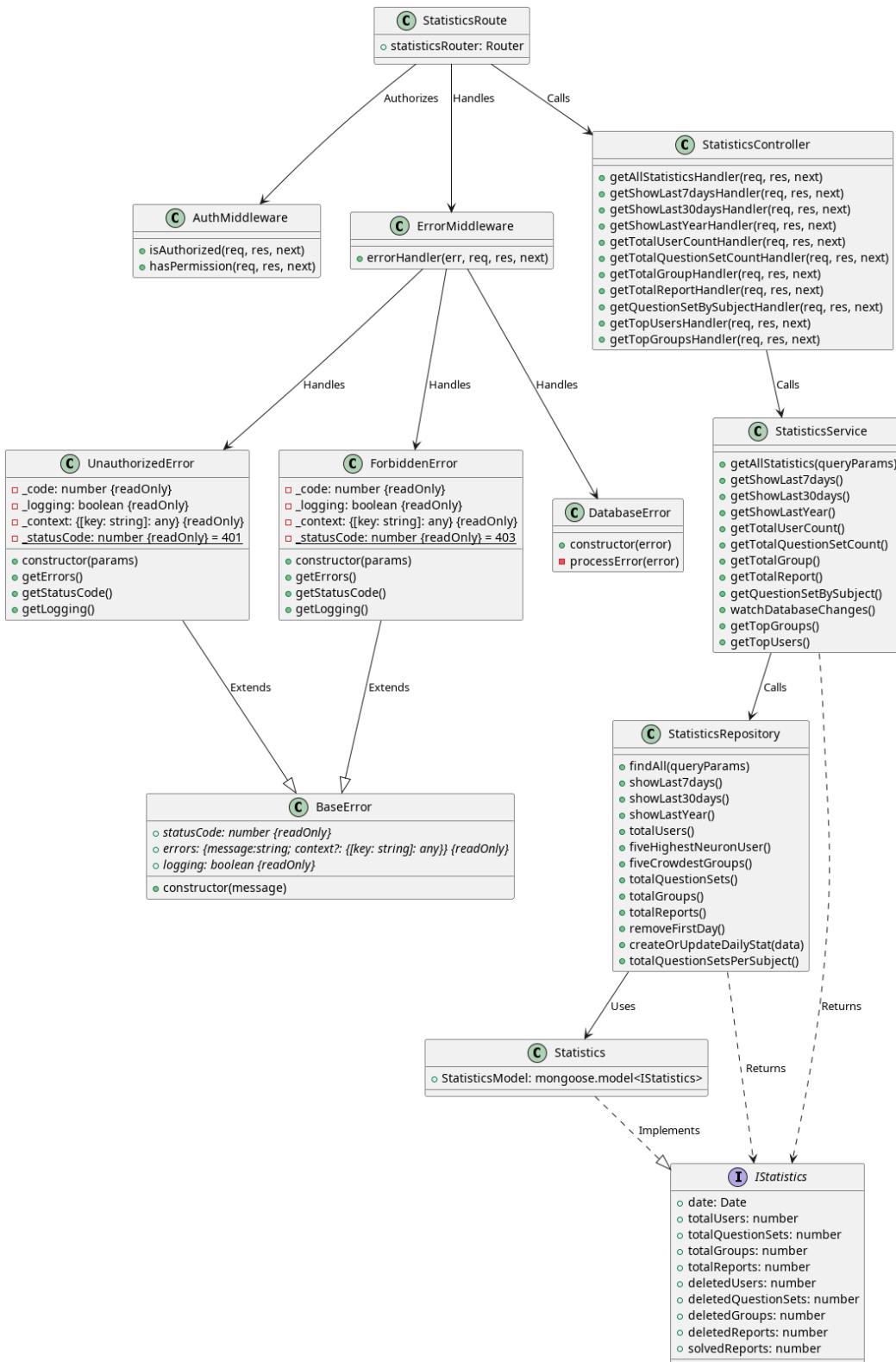


Figure 3.16.1.1.2: Get total question sets²³²

²³² [statistics.png](#)

3.16.1.1.3 Get total groups

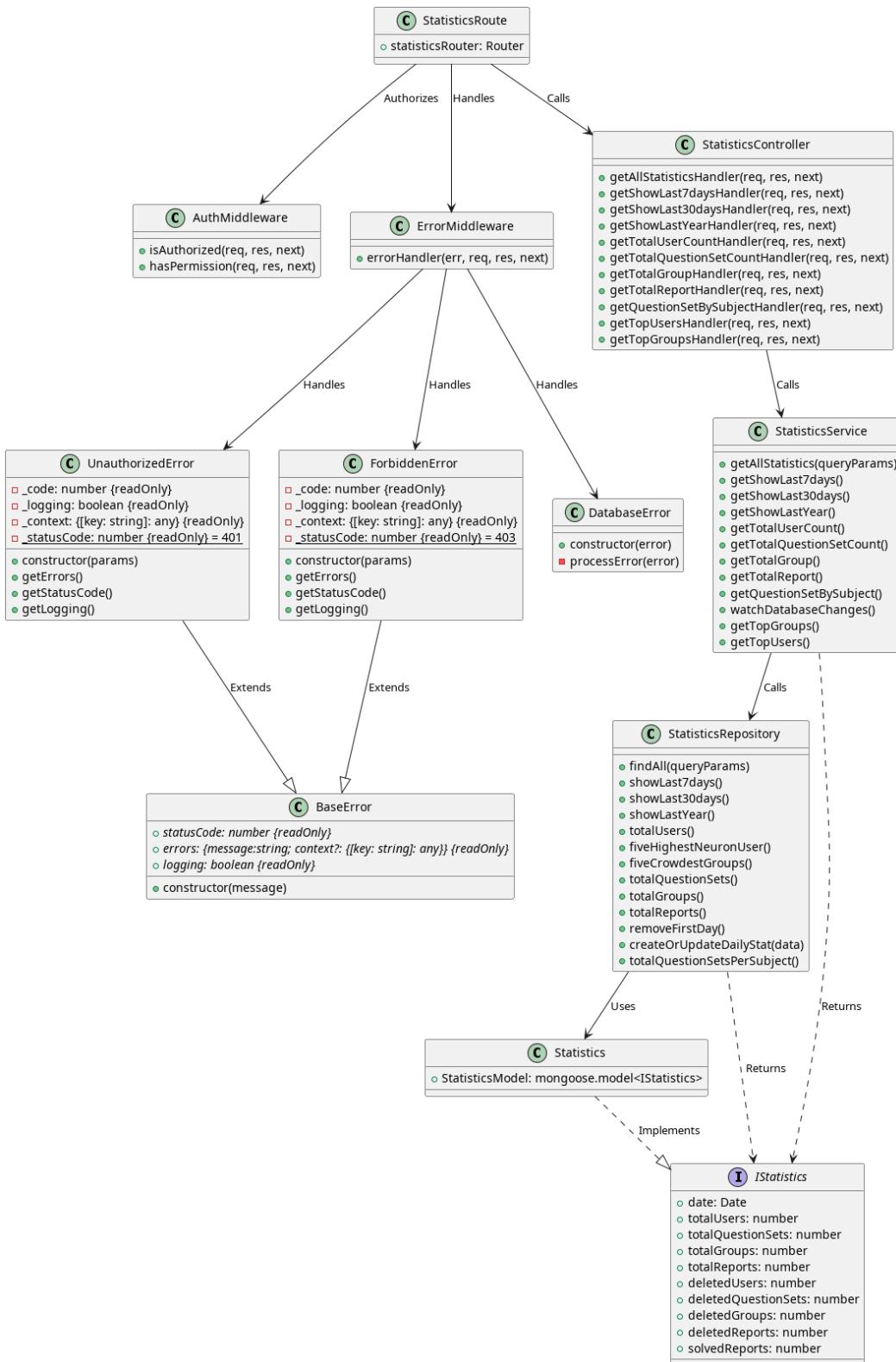


Figure 3.16.1.1.3: Get total groups²³³

²³³ [statistics.png](#)

3.16.1.1.4 Get total reports

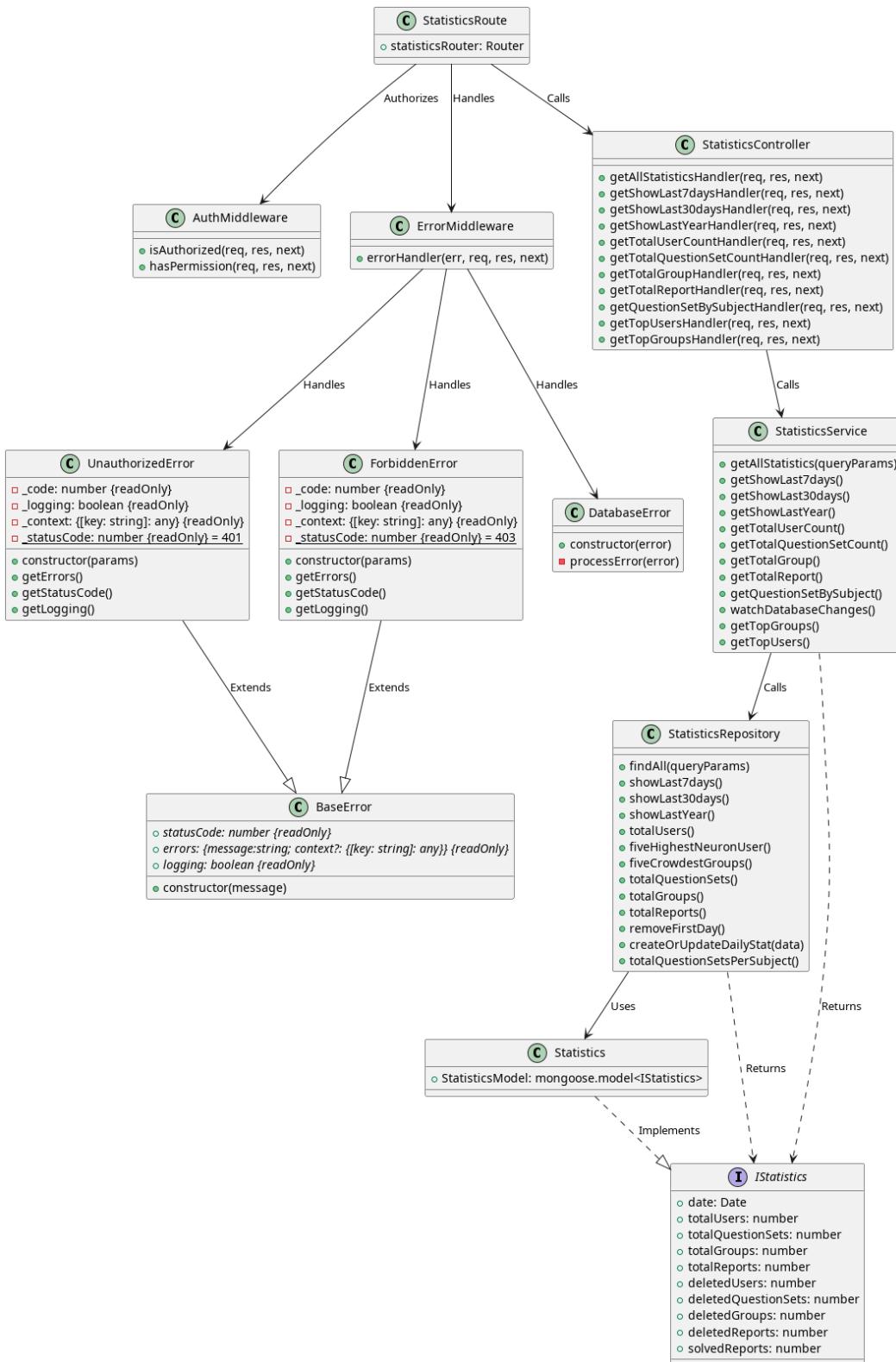


Figure 3.16.1.1.4: Get total report²³⁴

²³⁴ [statistics.png](#)

3.16.1.2 Sequence diagram

3.16.1.2.1 Get total users

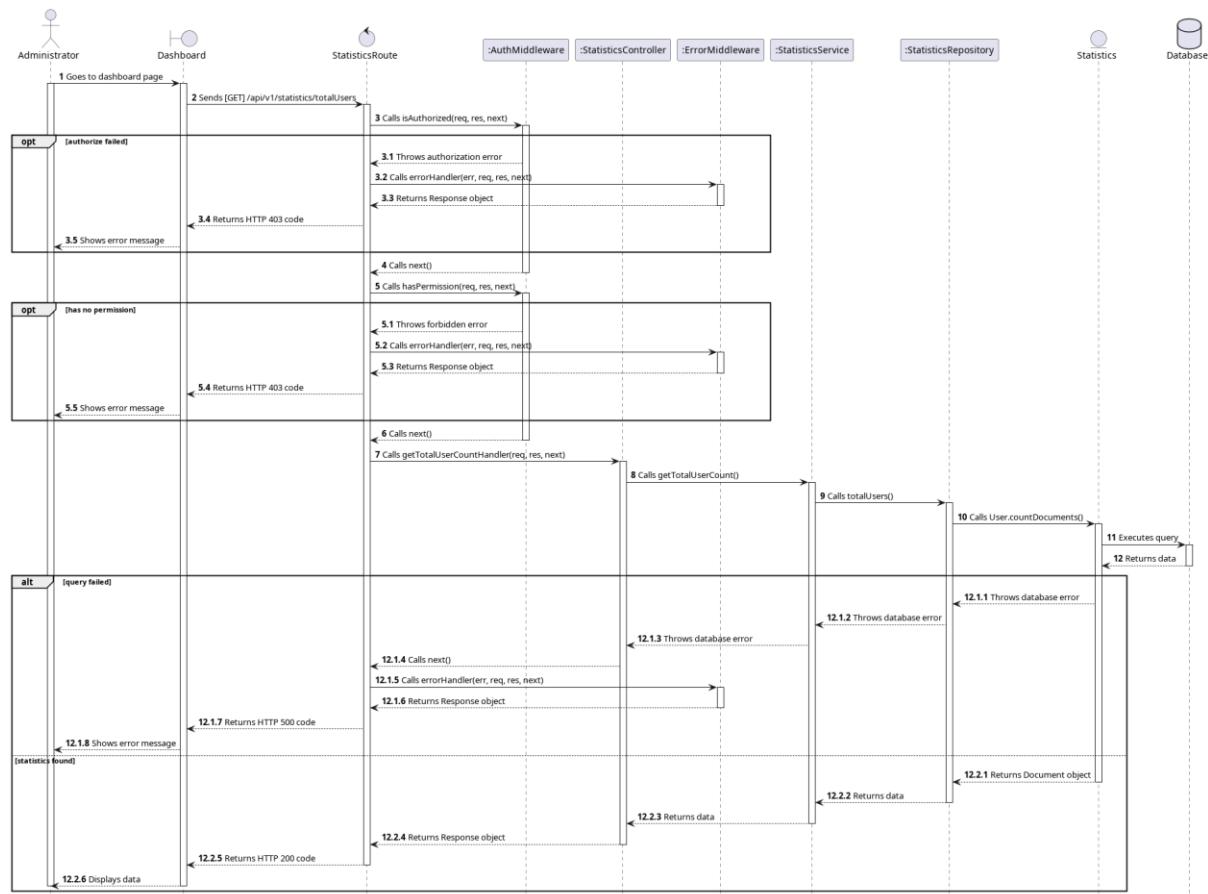


Figure 3.16.1.2.1: Get total users²³⁵

235 [get all user.png](#)

3.16.1.2.2 Get total question sets

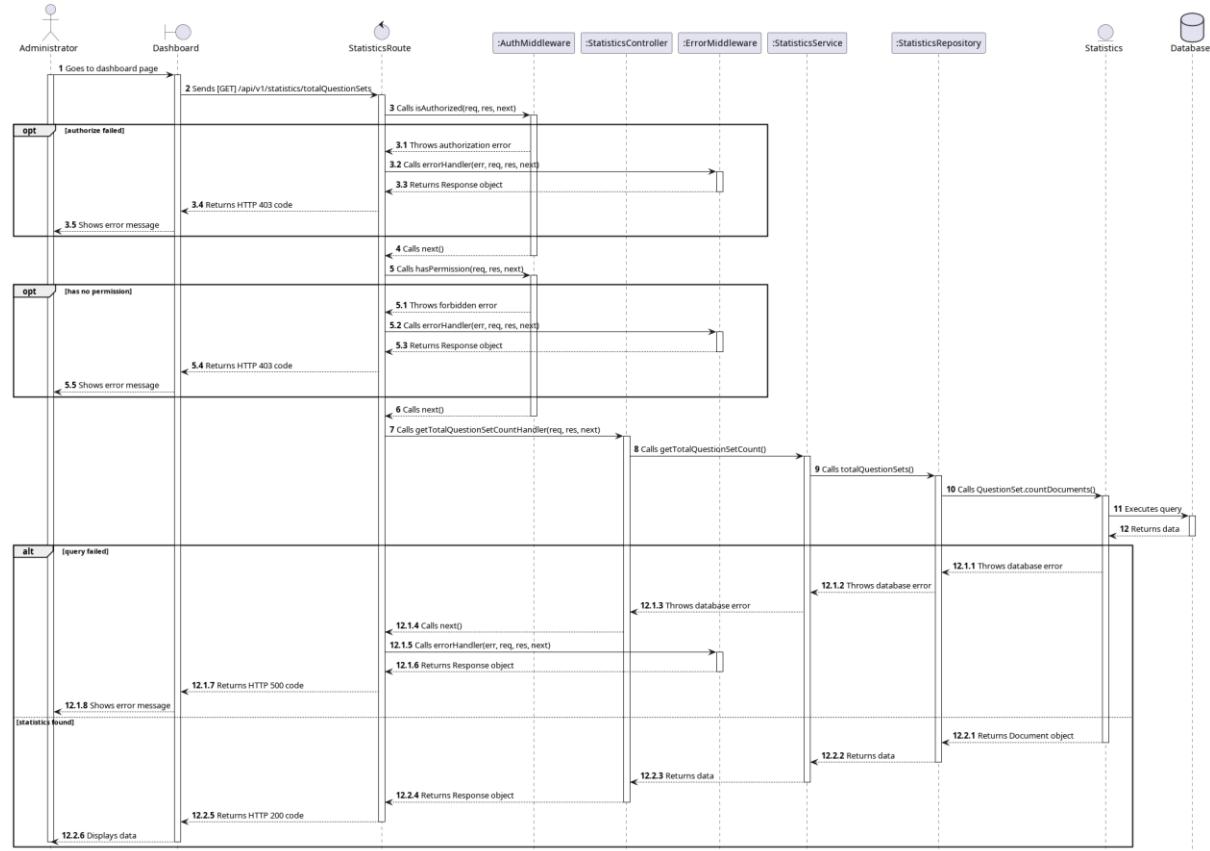


Figure 3.16.1.2.2: Get total question sets²³⁶

²³⁶ [get all question set.png](#)

3.16.1.2.3 Get total groups

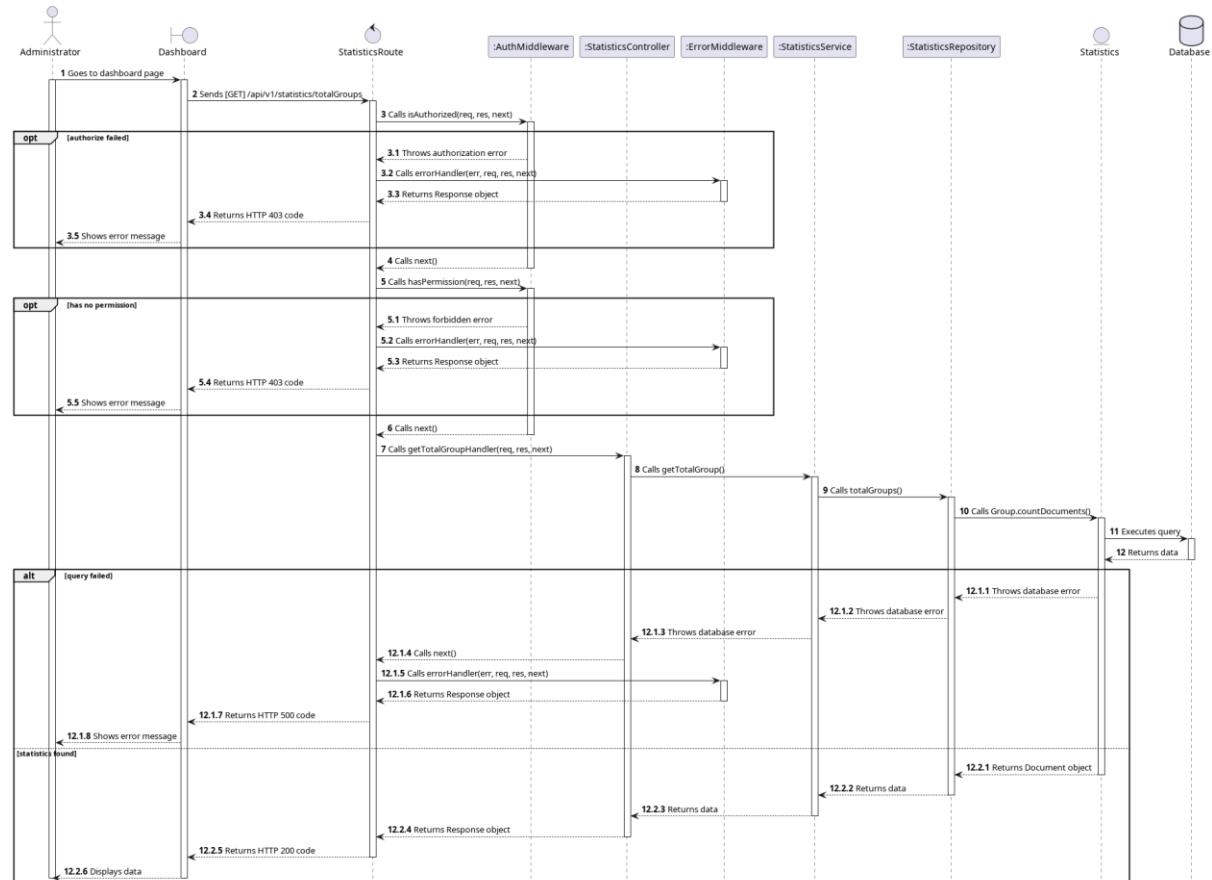


Figure 3.16.1.2.3: Get total groups²³⁷

²³⁷ [get all groups.png](#)

3.16.1.2.4 Get total reports

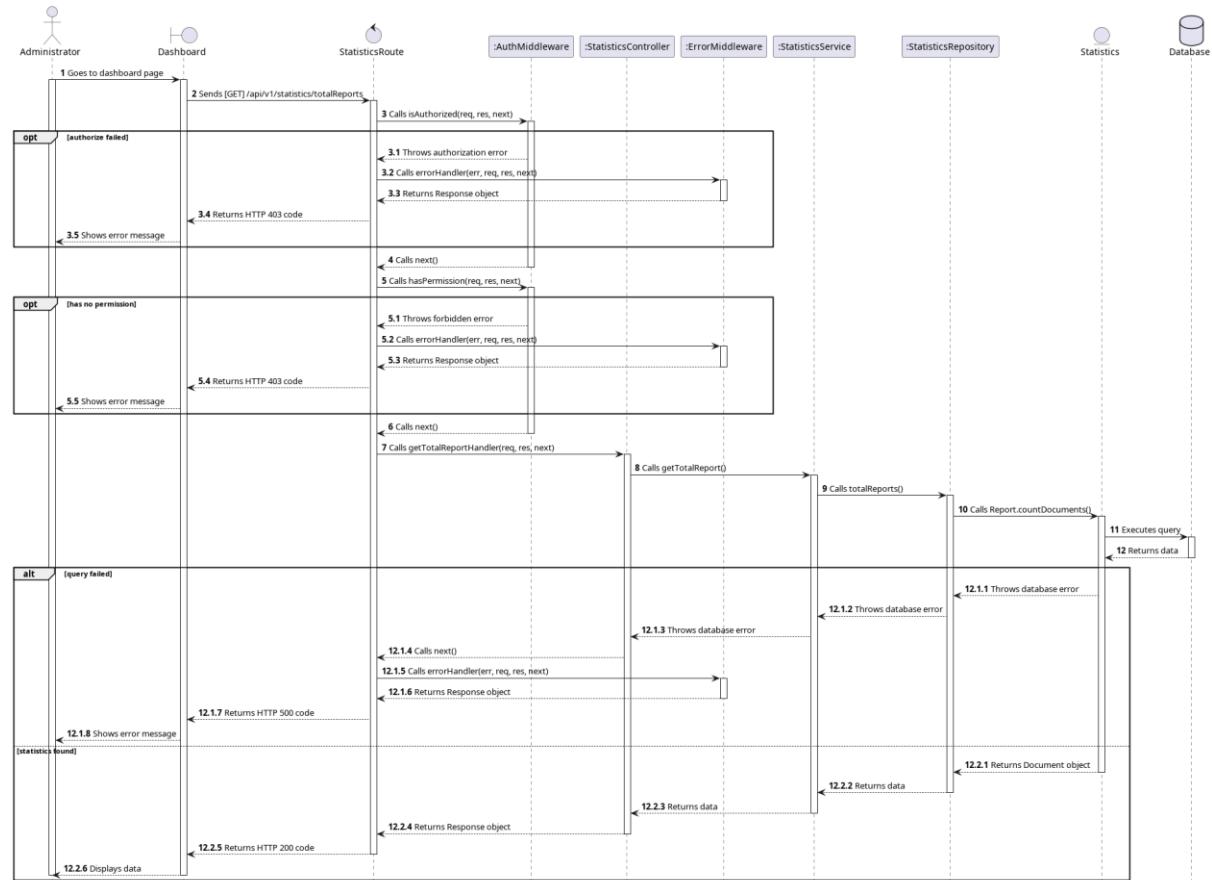


Figure 3.16.1.2.4: Get total reports²³⁸

²³⁸ [get all report.png](#)

3.16.2 Filter statistics

3.16.2.1 Class diagram

3.16.2.1.1 Filter from last week

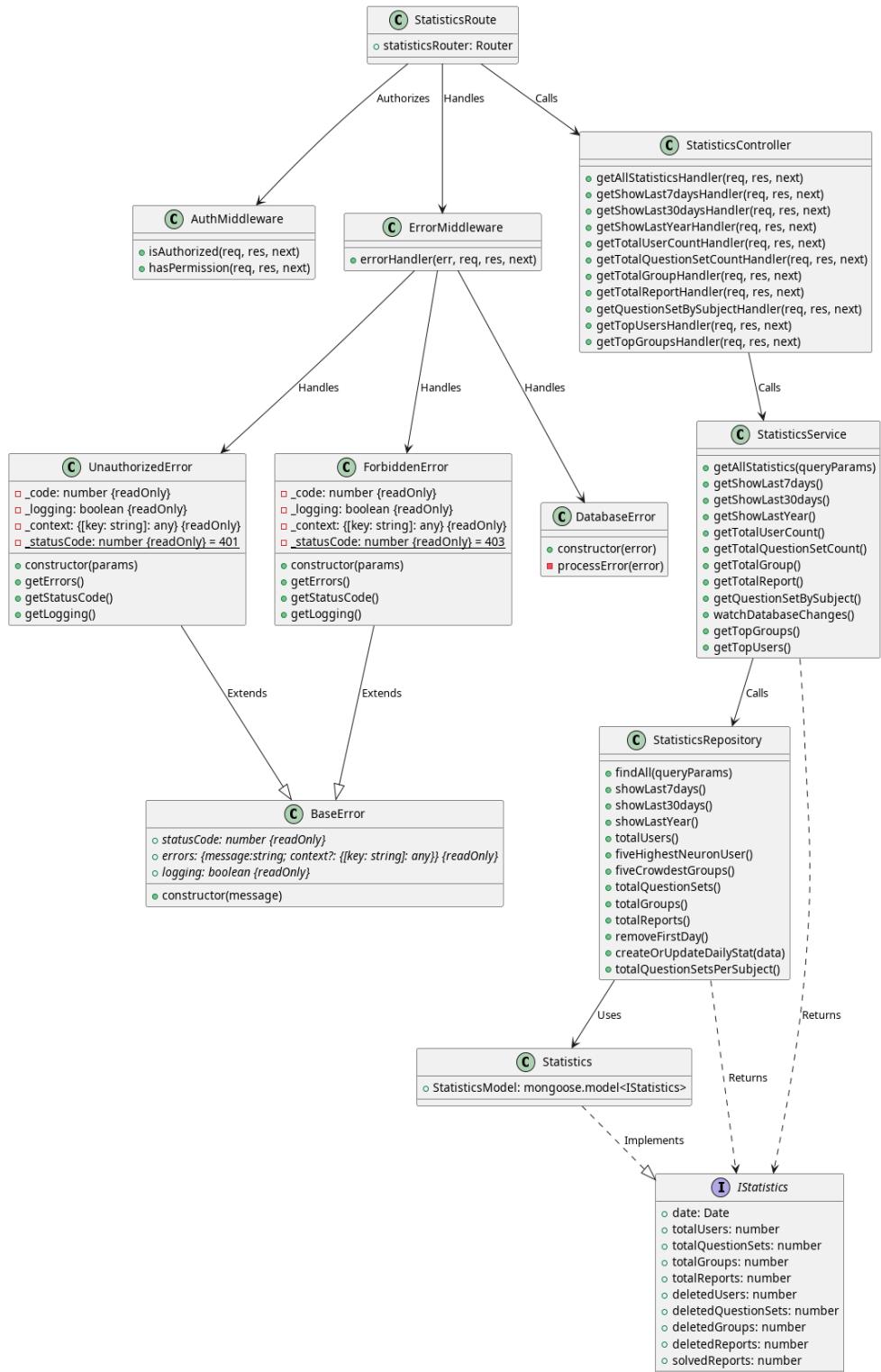
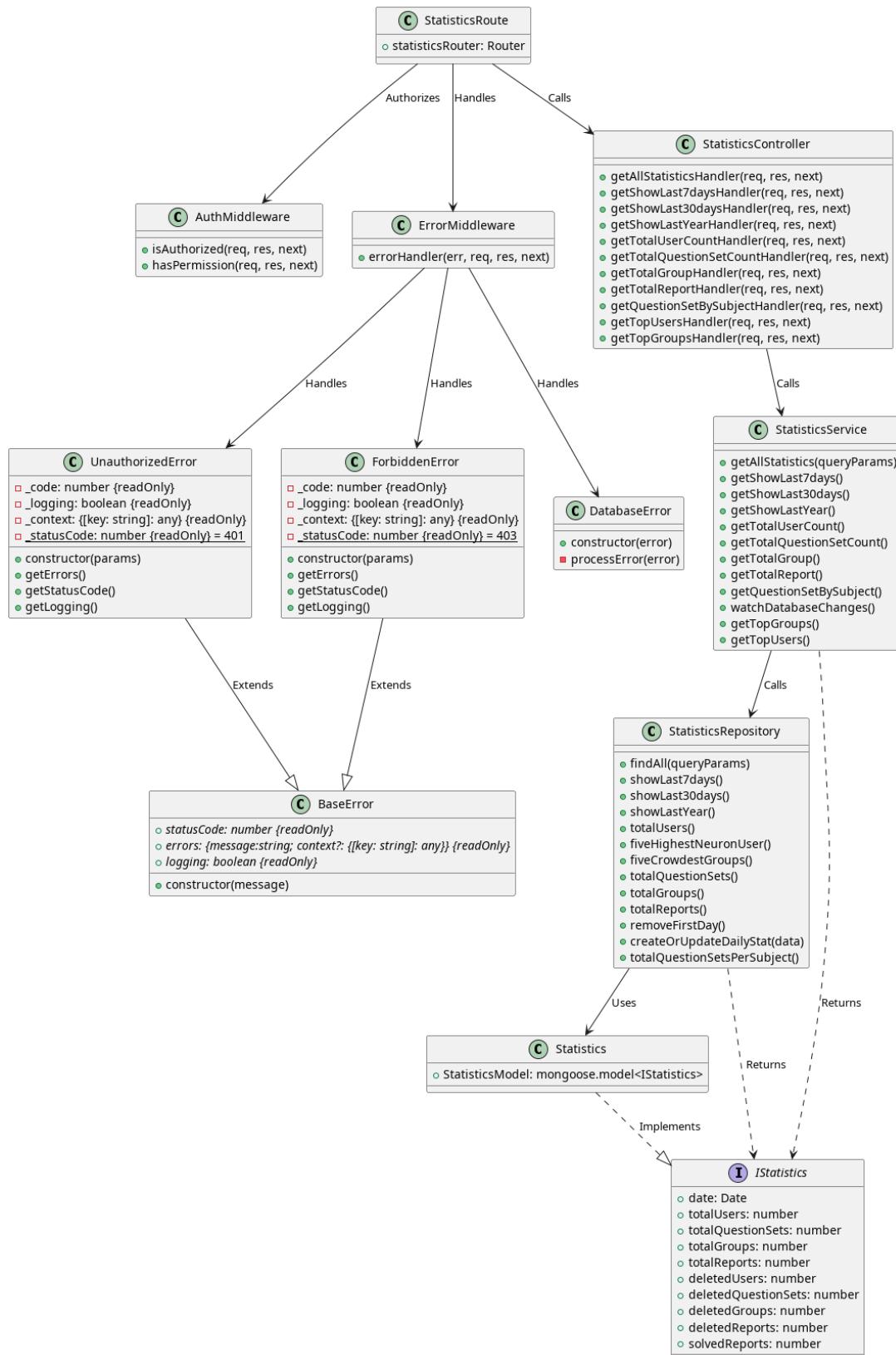


Figure 3.16.2.1.1: Filter from last week²³⁹

²³⁹ [statistics.png](#)

3.16.2.1.2 Filter from last month



*Figure 3.16.2.1.2: Filter from last month*²⁴⁰

240 statistics.png

3.16.2.1.3 Filter from last year

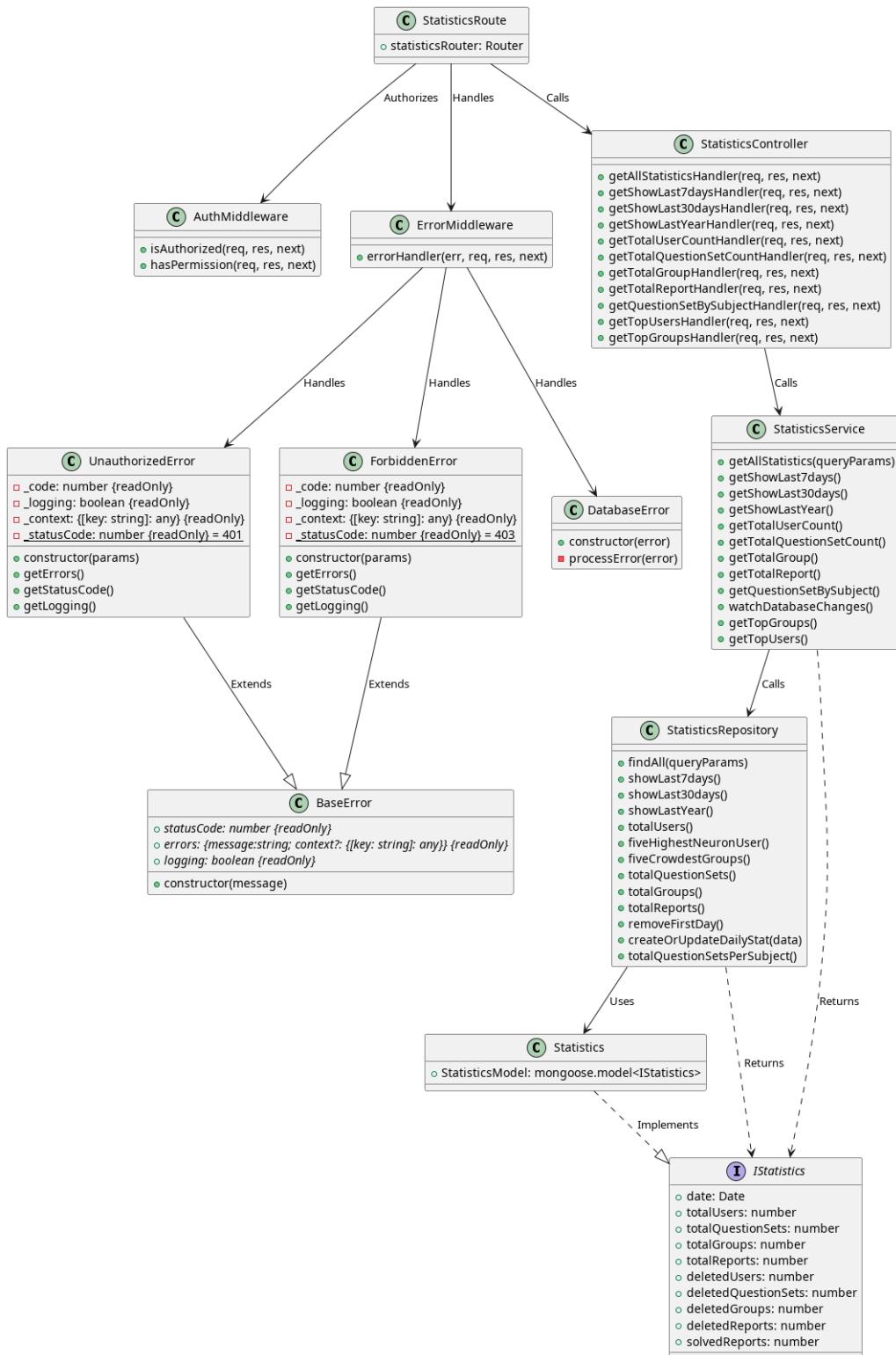


Figure 3.16.2.1.3: Filter from last year²⁴¹

²⁴¹ [statistics.png](#)

3.16.2.2 Sequence diagram

3.16.2.2.1 Filter from last week

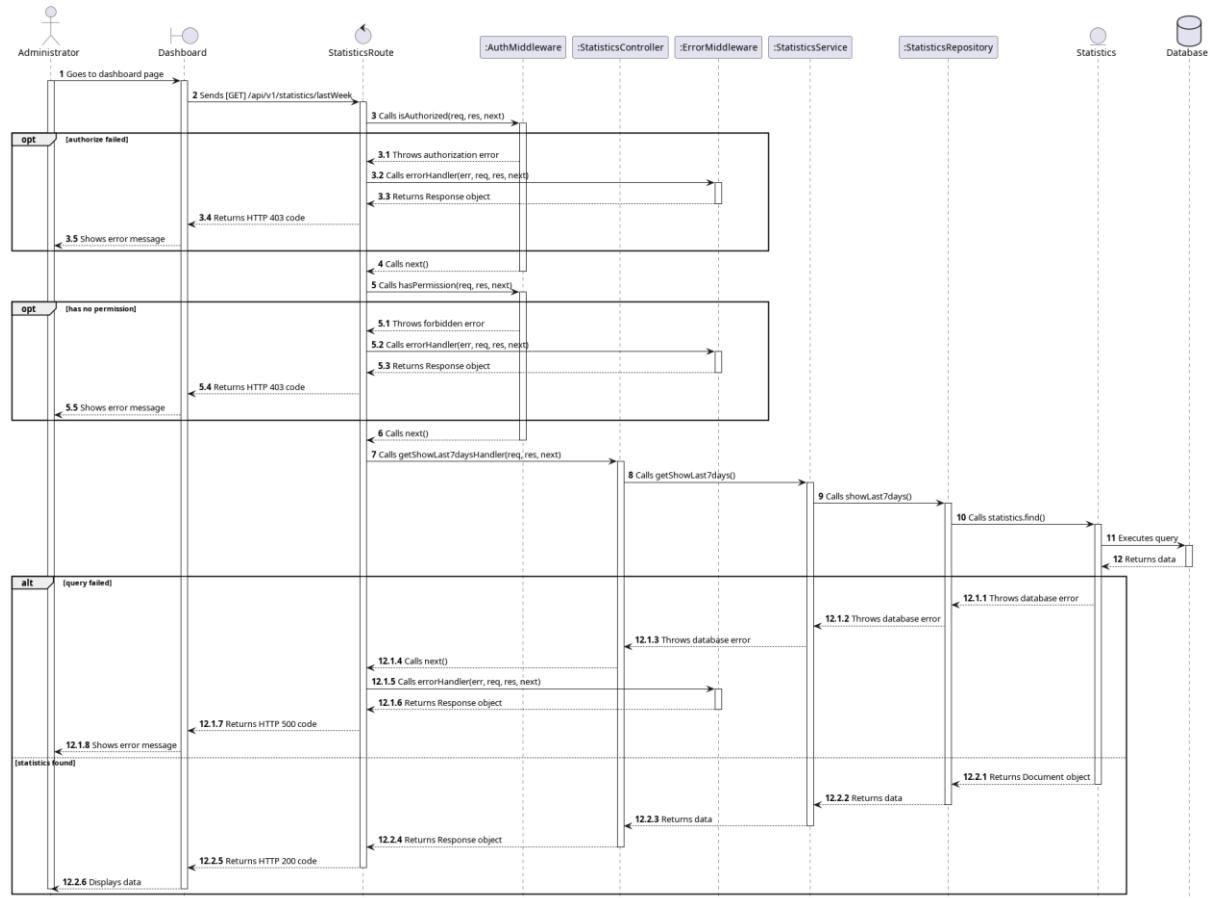


Figure 3.16.2.2.1: Filter from last week²⁴²

²⁴² [view last 7 days.png](#)

3.16.2.2.2 Filter from last month

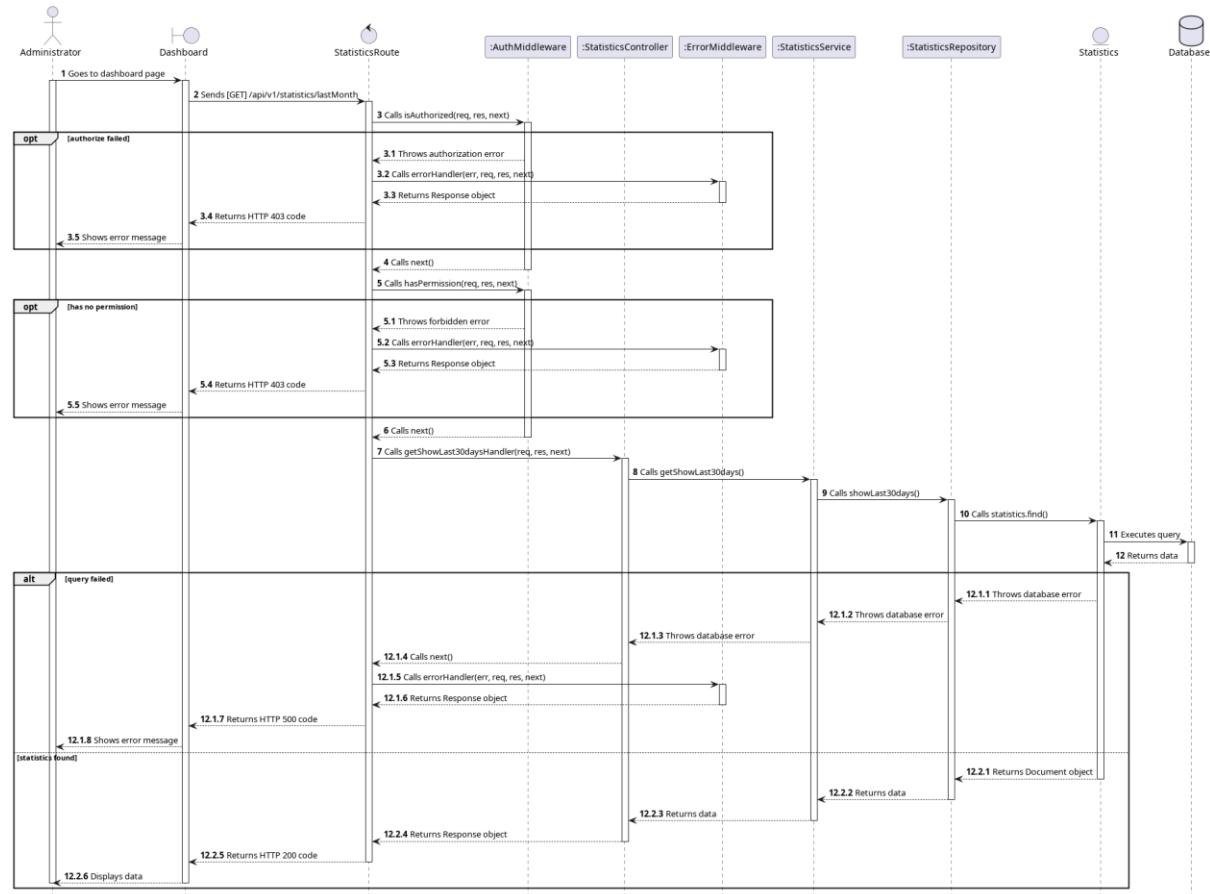


Figure 3.16.2.2.2: Filter from last month²⁴³

243 [view last month.png](#)

3.16.2.2.3 Filter from last year

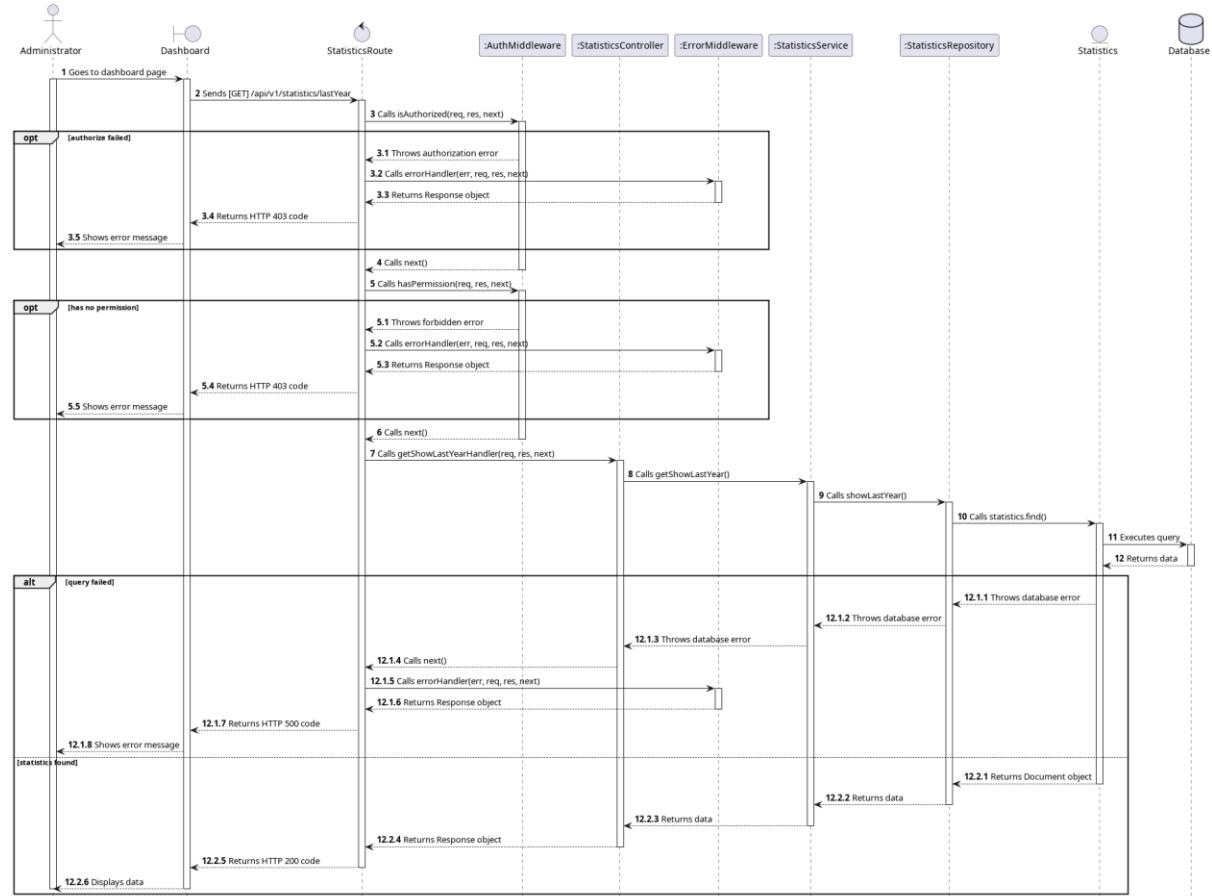


Figure 3.16.2.2.3: Filter from last year²⁴⁴

244 [view last year.png](#)

3.17 Additional Features

3.17.1 Use Pomodoro timer

This feature is client-only, so class diagram will not be shown

3.17.1.1 Sequence diagram

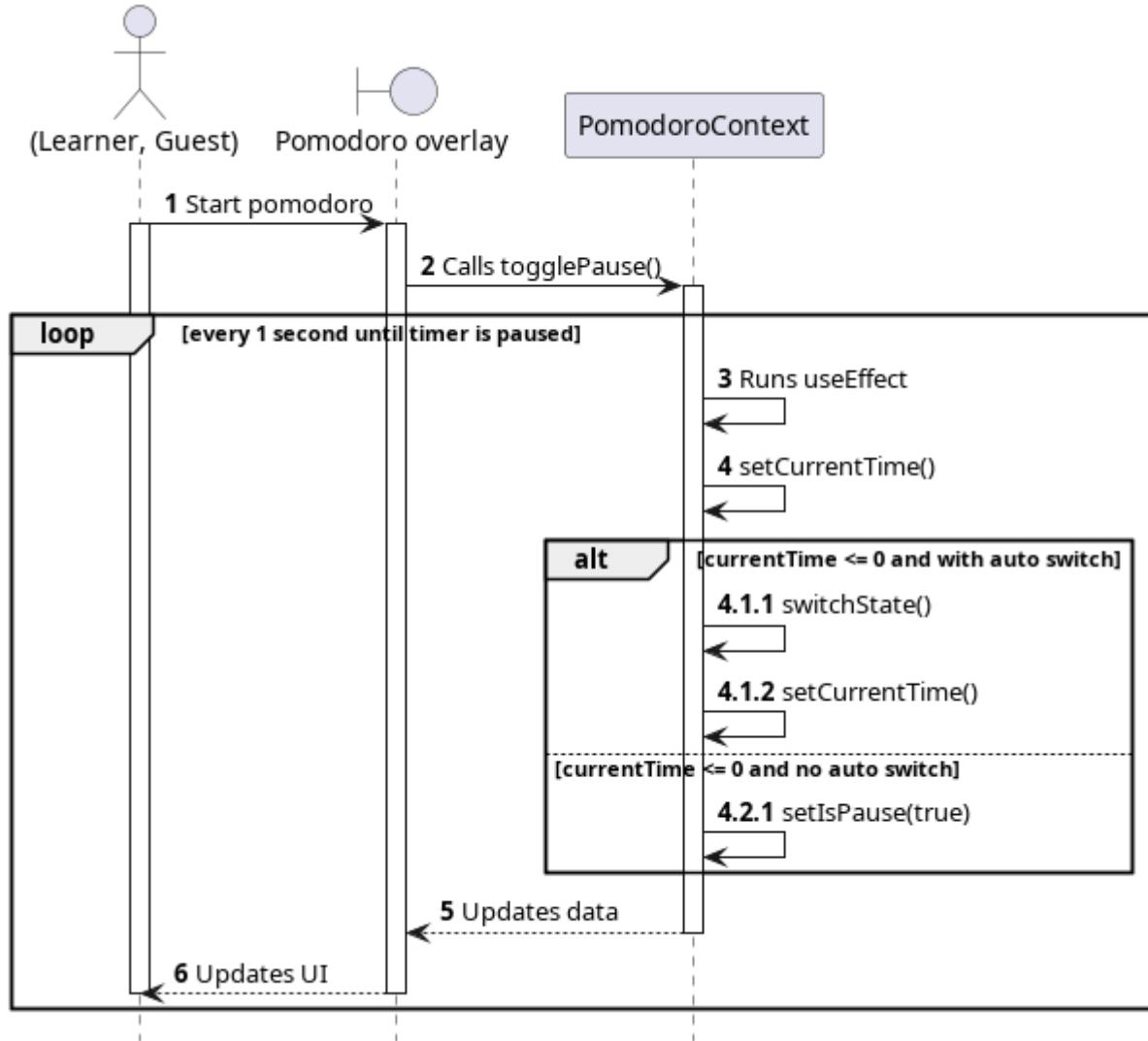


Figure 3.17.1.1: Use Pomodoro timer²⁴⁵

²⁴⁵ pomodoro.png

3.17.2 View achievements

3.17.2.1 Class diagram

3.17.2.1.1 View achievement

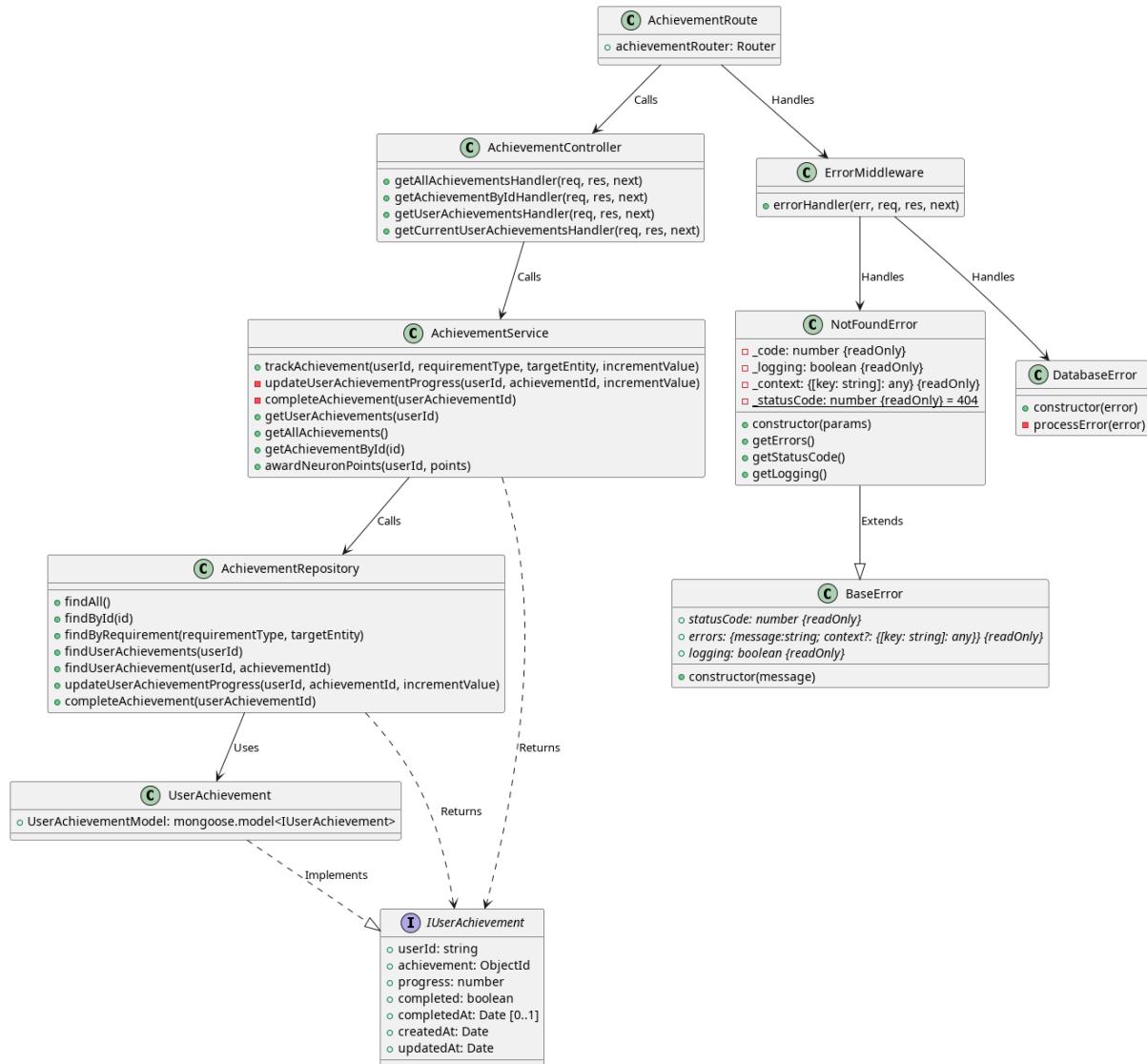


Figure 3.17.2.1.1: View achievement list²⁴⁶

²⁴⁶ [view personal achievements.png](#)

3.17.2.2 Sequence diagram

3.17.2.2.1 View achievement list

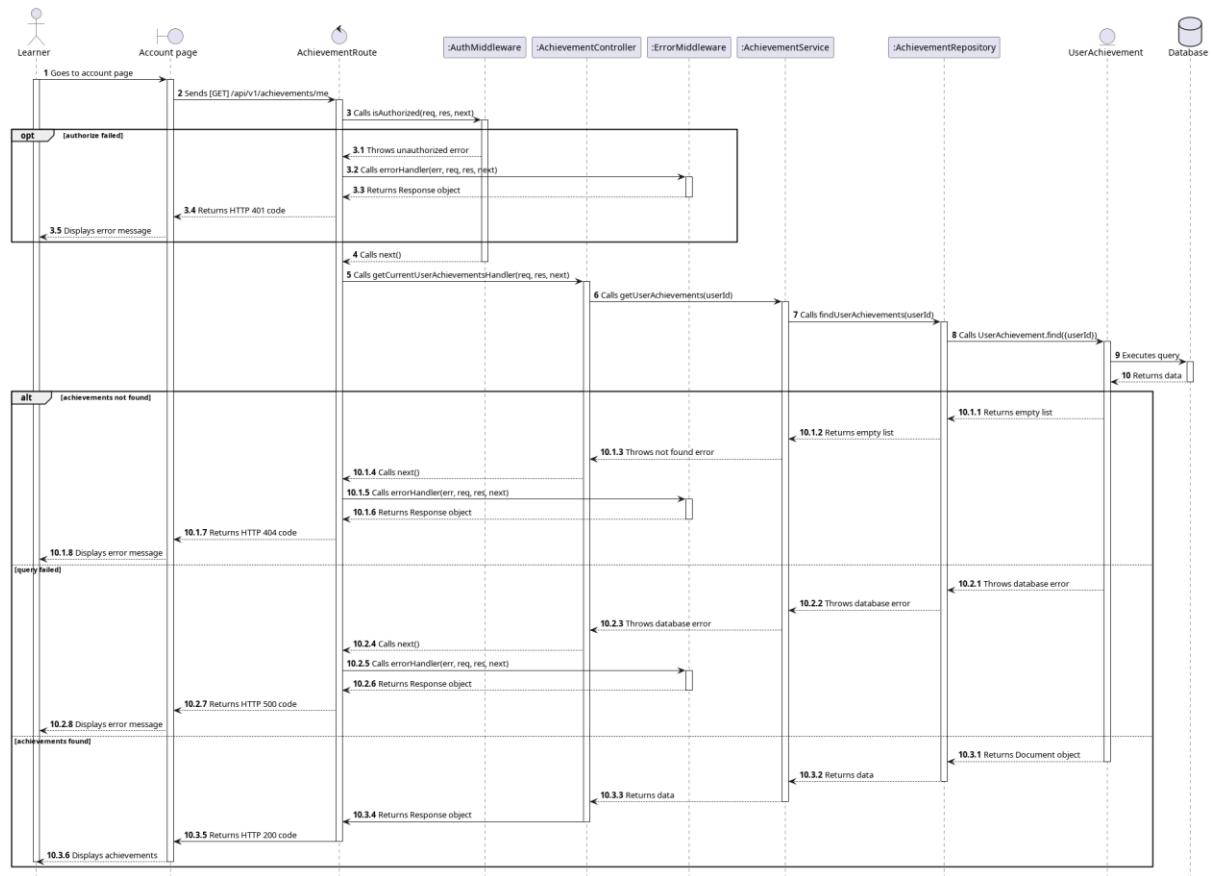


Figure 3.17.2.2.1: View achievement list²⁴⁷

²⁴⁷ [view personal achievements.png](#)

3.17.3 View quote

This feature is client-only, so class diagram will not be shown

3.17.3.1 Sequence diagram

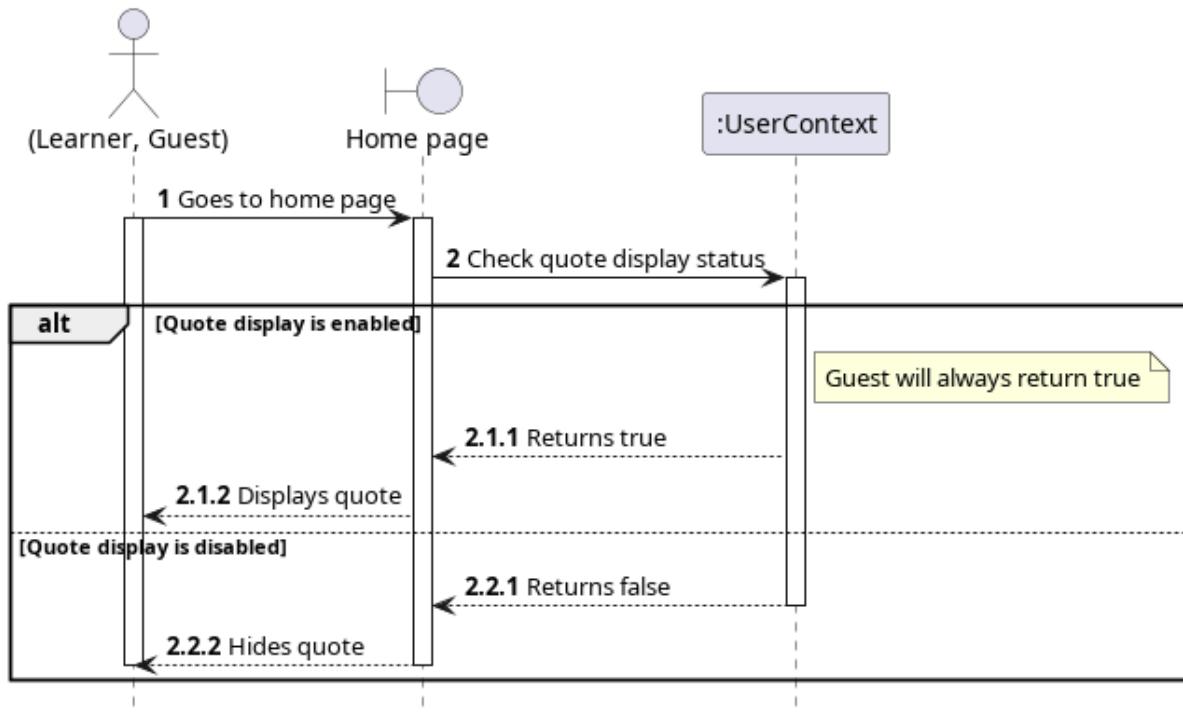


Figure 3.17.3.1: View quote²⁴⁸

²⁴⁸ [view quotes.png](#)

4. Class Specification

4.1 Clerk Classes

- Clerk class

No.	Attributes/ Methods	Scope	Description
01	user: ClerkUser [0..1]	public	Holds the currently active User object. If the session is null or undefined, the user field will match.
02	loaded: boolean	public	A getter to see if the Clerk object is ready for use or not
03	session: Session [0..1]	public	The currently active Session, which is guaranteed to be one of the sessions in Client.sessions
04	publishableKey: string [0..1]	public	The Clerk Publishable Key for the project instance
05	useAuth(): UseAuthReturn	public	A method that provides access to the current user's authentication state and methods to manage the active session in your Vue application.
06	useUser(): UseUserReturn	public	A method that provides access to the current user's User object, which contains all the data for a single user in your application and provides methods to manage their account. This composable also allows checking if the user is signed in and if Clerk has loaded and initialized.
07	authenticateWithGoogleOneTap(props?: AuthenticateWithGoogleOneTapParams): Promise<SignInResource SignUpResource>	public	Authenticates the user with a token generated from Google identity services. Also sets the user's current session to active.
08	handleGoogleOneTapCallback(signInOrUp: SignInResource SignUpResource, params: HandleOAuthCallbackParams, customNa vigate?: (to: string) => Promise<unknown>): Promise<unknown>	public	Completes a Google One Tap redirection flow started by authenticateWithGoogleOneTap(). Also calls Clerk.setActive()
09	setActive(params: SetActiveParams): Promise<void>	public	A method used to set the active session and/or organization

10	signOut(options?: SignOutOptions): Promise<void>	public	A method that delete the current client session
----	--------------------------------------------------	--------	-------------------------------------------------

- ClerkService class

No.	Attributes/ Methods	Scope	Description
01	user: ClerkUser	public	An attribute that use ClerkUser class
02	clerk: User	public	An attribute that use Clerk class
03	signIn: SignIn	public	An attribute that use SignIn class
04	signUp: SignUp	public	An attribute that use SignUp class

- ClerkUser class

No.	Attributes/ Methods	Scope	Description
01	id: string	public	Contains the user's unique identifier.
02	createdAt: Date	public	Contains the date when the user was first created.
03	deleteSelfEnabled: boolean	public	Indicating whether the user is able to delete their own account.
04	emailAddresses: string[]	public	An array of all the EmailAddress objects associated with the user. Includes the primary.
05	primaryEmailAddressId: string [0..1]	public	Contains the ID for the EmailAddress that the user has set as primary.
06	externalAccounts: string[]	public	An array of all the ExternalAccount objects associated with the user via OAuth. This includes both verified & unverified external accounts.
07	externalId: string [0..1]	public	The user's ID as used in your external systems. Must be unique across your instance.
08	username: string	public	Contains the user's username.

09	firstName: string [0..1]	public	Contains the user's first name.
10	lastName: string [0..1]	public	Contains the user's last name.
11	fullName: string [0..1]	public	Contains the user's full name.
12	hasImage: boolean	public	Indicates whether the user has uploaded an image or one was copied from OAuth. Returns false if Clerk is displaying an avatar for the user.
13	hasVerifiedEmailAddress: boolean	public	Indicates whether the user has verified an email address.
14	imageUrl: string	public	Contains the default avatar or user's uploaded profile image
15	lastSignInAt: Date [0..1]	public	The date when the user last signed in. null if the user has never signed in.
16	privateMetaData: UserPrivateMetaData	public	Metadata that can be read and set only from the Backend API
17	publicMetaData: UserPublicMetaData	public	Metadata that can be read from the Frontend API and Backend API and can be set only from the Backend API.
18	verifiedExternalAccounts: string[]	public	An array of all the ExternalAccount objects associated with the user via OAuth that are verified.
19	update(params: UpdateUserParams): Promise<User>	public	A method that updates the user's attributes
20	updatePassword(params: UpdateUserPasswordParams): Promise<User>	public	A method that updates the user's password
21	delete(): Promise<void>	public	A method that deletes the current user.
22	createExternalAccount(params: CreateExternalAccountParams): Promise<ExternalAccount>	public	A method that adds an external account for the user. A new ExternalAccount will be created and associated with the user

- SignIn class

No.	Attributes/ Methods	Scope	Description
01	status: SignInStatus	public	Contains the current status of the sign-in
02	supportedIdentifiers: SignInIdentifiers[]	public	Array of all the authentication identifiers that are supported for this sign in.
03	identifier: string [0..1]	public	The authentication identifier value for the current sign-in.
04	supportFirstFactors: SignInFirstFactor[]	public	Array of the first factors that are supported in the current sign-in. Each factor contains information about the verification strategy that can be used
05	firstFactorVerification: Verification	public	The state of the verification process for the selected first factor.
06	userData: UserData	public	An object containing information about the user of the current sign-in. This property is populated only once an identifier is given to the SignIn object.
07	createdSessionId: string	public	Contains the session that was created upon completion of the current sign-in. The value of this property is null if the sign-in status is not 'complete'.
08	attemptFirstFactor(params: AttemptFirstFactorParams): Promise<SignIn>	public	A method that attempts to complete the first factor verification process. This is a required step in order to complete a sign in, as users should be verified at least by one factor of authentication.
09	authenticateWithRedirect(params: AuthenticateWithRedirectParams): Promise<void>	public	A method that signs in a user via a Single Sign On (SSO) connection, such as OAuth or SAML, where an external account is used for verifying the user's identity.
10	create(params: SignInCreateParams): Promise<SignIn>	public	A method that creates and returns a new SignIn instance initialized with the provided parameters. The instance maintains the sign-in lifecycle state through its status property, which updates as the authentication flow progresses.
11	prepareFirstFactor(params: PrepareFirstFactorParams): Promise<SignIn>	public	A method that begins the first factor verification process. This is a required step in order to complete a sign in, as users should be verified at least by one factor of authentication.

12	<code>resetPassword(params: ResetPasswordParams): Promise<SignIn></code>	public	A method that resets a user's password
----	--------------------------------------------------------------------------------	--------	----------------------------------------

- SignUp class

No.	Attributes/ Methods	Scope	Description
01	<code>id: string [0..1]</code>	public	Contains the unique identifier of the current sign-up.
02	<code>status: string [0..1]</code>	public	Contains the status of the current sign-up
03	<code>requiredFields: string[]</code>	public	An array of all the required fields that need to be supplied and verified in order for this sign-up to be marked as complete and converted into a user.
04	<code>optionalFields: string[]</code>	public	An array of all the fields that can be supplied to the sign-up, but their absence does not prevent the sign-up from being marked as complete.
05	<code>missingFields: string[]</code>	public	An array of all the fields whose values are not supplied yet but they are mandatory in order for a sign-up to be marked as complete.
06	<code>unverifiedFields: string[]</code>	public	An array of all the fields whose values have been supplied, but they need additional verification in order for them to be accepted
07	<code>verifications: SignUpVerifications</code>	public	An object that contains information about all the verifications that are in-flight.
08	<code>username: string [0..1]</code>	public	Contains the username supplied to the current sign-up.
09	<code>emailAddress: string [0..1]</code>	public	Contains the email address supplied to the current sign-up
10	<code>hasPassword: boolean</code>	public	Contains the value of this attribute is true if a password was supplied to the current sign-up.
11	<code>firstName: string [0..1]</code>	public	Contains the first name supplied to the current sign-up.
12	<code>lastName: string [0..1]</code>	public	Contains the last name supplied to the current sign-up.
13	<code>createdSessionId: string [0..1]</code>	public	The identifier of the newly-created session. This attribute is populated only when the sign-up is

			complete.
14	createdUserId: string [0..1]	public	The identifier of the newly-created user. This attribute is populated only when the sign-up is complete.
15	abandonAt: number [0..1]	public	The numerical time when the sign-up was abandoned by the user.
16	attemptEmailAddressVerification(params: AttemptEmailAddressVerificationParams): Promise<SignUpResource>	public	A method that attempts to verify an email address by validating the one-time verification code provided by the user against the code sent during the preparation verification step.
17	attemptVerification(params: AttemptVerificationParams): Promise<SignUp>	public	A method that attempts to complete a pending verification process for the specified verification strategy.
18	authenticateWithRedirect(params: AuthenticateWithRedirectParams): Promise<void>	public	A method that signs up a user via a Single Sign On (SSO) connection, such as OAuth or SAML, where an external account is used for verifying the user's identity.
19	create(params: SignUpCreateParams): Promise<SignUpResource>	public	A method that returns a new SignUp object based on the params you pass to it, stores the sign-up lifecycle state in the status property, and deactivates any existing sign-up process the client may already have in progress. Use this method to initiate a new sign-up process.
20	prepareEmailAddressVerification(params?: PrepareEmailAddressVerificationParams): Promise<SignUpResource>	public	A method that initiates an email verification process by sending a one-time verification code to the email address associated with the current sign-up attempt.
21	prepareVerification(params: PrepareVerificationParams): Promise<SignUpResource>	public	A method that initiates the verification process for a field that requires validation during sign-up.

4.2 Route Classes

- CommentRoute class

No.	Attributes/ Methods	Scope	Description
01	commentRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

- FieldRoute class

No.	Attributes/ Methods	Scope	Description
01	fieldRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

- GoalRoute class

No.	Attributes/ Methods	Scope	Description
01	goalRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

- GroupRoute class

No.	Attributes/ Methods	Scope	Description
01	groupRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

- ProgressRoute class

No.	Attributes/ Methods	Scope	Description
01	progressRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

- QuestionRoute class

No.	Attributes/ Methods	Scope	Description
01	questionRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

- QuestionSetRoute class

No.	Attributes/ Methods	Scope	Description
01	questionSetRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

- QuizRoute class

No.	Attributes/ Methods	Scope	Description
01	quizRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

- ReportRoute class

No.	Attributes/ Methods	Scope	Description
01	reportRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

- SubjectRoute class

No.	Attributes/ Methods	Scope	Description
01	subjectRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

- UserRoute class

No.	Attributes/ Methods	Scope	Description
01	userRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

- UserFollow class

No.	Attributes/ Methods	Scope	Description
01	userFollowRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

- AchievementRoute class

No.	Attributes/ Methods	Scope	Description
01	achievementRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

- StatisticsRoute class

No.	Attributes/ Methods	Scope	Description
01	statisticsRouter: Router	public	Contains methods that listens for requests that match the specified route(s) and method(s)

4.3 Models

- Achievement class

No.	Attributes/ Methods	Scope	Description
01	AchievementModel: mongoose.model<IAchievement>	public	Contains MongoDB schema of achievement

- Comment class

No.	Attributes/ Methods	Scope	Description
01	CommentModel: mongoose.model<IComment>	public	Contains MongoDB schema of comment

- CommentVote class

No.	Attributes/ Methods	Scope	Description
01	CommentVoteModel: mongoose.model<ICommentVote>	public	Contains MongoDB schema of user vote for a comment

- Field class

No.	Attributes/ Methods	Scope	Description
01	FieldModal: mongoose.model<IField>	public	Contains MongoDB schema of field

- Goal class

No.	Attributes/ Methods	Scope	Description
01	GoalModel: mongoose.model<IGoal>	public	Contains MongoDB schema of goal

- Invitation class

No.	Attributes/ Methods	Scope	Description
01	InvitationModel: mongoose.model<IInvitation>	public	Contains MongoDB schema of invitation

- Group class

No.	Attributes/ Methods	Scope	Description
01	GroupModel: mongoose.model<IGroup>	public	Contains MongoDB schema of group

- GroupMember class

No.	Attributes/ Methods	Scope	Description
01	GroupMemberModel: mongoose.model<IGroupMember>	public	Contains MongoDB schema of group member

- GroupQuestionSet class

No.	Attributes/ Methods	Scope	Description
01	GroupQuestionModel: mongoose.model<IGroupQuestionSet>	public	Contains MongoDB schema of group shared question set

- Question class

No.	Attributes/ Methods	Scope	Description
01	QuestionModel: mongoose.model<IQuestion>	public	Contains MongoDB schema of question

- QuestionProgress class

No.	Attributes/ Methods	Scope	Description
01	QuestionProgressModel: mongoose.model<IQuestionProgress>	public	Contains MongoDB schema of question progress

- QuestionSet class

No.	Attributes/ Methods	Scope	Description
01	QuestionSetModel: mongoose.model<IQuestionSet>	public	Contains MongoDB schema of question set

- QuestionSetProgress class

No.	Attributes/ Methods	Scope	Description
01	QuestionSetProgressModel: mongoose.model<IQuestionSetProgress>	public	Contains MongoDB schema of question set progress

- QuizResult class

No.	Attributes/ Methods	Scope	Description
01	QuizResultModel: mongoose.model<IQuizResult>	public	Contains MongoDB schema of quiz result

- Report class

No.	Attributes/ Methods	Scope	Description
01	ReportModel: mongoose.model<IReport>	public	Contains MongoDB schema of report

- Statistics class

No.	Attributes/ Methods	Scope	Description
01	StatisticsModel: mongoose.model<IStatistics>	public	Contains MongoDB schema of statistics

- Subject class

No.	Attributes/ Methods	Scope	Description
01	SubjectModel: mongoose.model<ISubject>	public	Contains MongoDB schema of subject

- User class

No.	Attributes/ Methods	Scope	Description
01	UserModel: mongoose.model<IUser>	public	Contains MongoDB schema of user

- UserAchievement class

No.	Attributes/ Methods	Scope	Description
01	UserAchievementModel: mongoose.model<IUserAchievement>	public	Contains MongoDB schema of achievement progress of a user

- UserFollow class

No.	Attributes/ Methods	Scope	Description
01	UserFollowModel: mongoose.model<IUserFollow>	public	Contains MongoDB schema of follow status of a user

- QuizResultQuestion class

No.	Attributes/ Methods	Scope	Description
01	QuizResultQuestionModel: mongoose.model<IQuizResultQuestion>	public	Contains MongoDB schema of follow status of a quiz result option

4.4 Interfaces

- IAchievement class

No.	Attributes/ Methods	Scope	Description
01	title: string	public	Contains achievement title
02	description: string	public	Contains achievement description
03	requirements: { type: string threshold: number targetEntity: string }	public	Contains achievement requirement
04	image: object	public	Contains achievement image
05	category: string	public	Contains achievement category
06	createdAt: Date	public	Contains achievement create date
07	updatedAt: Date	public	Contains achievement update date

- IComment class

No.	Attributes/ Methods	Scope	Description
01	questionSetId: mongoose.Types.ObjectId	public	Contains comment original question set id
02	questionId: mongoose.Types.ObjectId	public	Contains comment original question id

03	parentCommentId: mongoose.Types.ObjectId	public	Contains comment parent id
04	user: string	public	Contains comment user id
05	comment: string	public	Contains comment content
06	upvote: number	public	Contains comment upvote
07	downvote: number	public	Contains comment downvote
08	priority: Priority	public	Contains comment priority
09	replies: mongoose.Types.ObjectId[]	public	Contains comment replies id as a list
10	status: Status	public	Contains comment status
11	isAccepted: boolean	public	Contains comment acceptance from question set author
12	createdAt: Date	public	Contains comment create date
13	updatedAt: Date	public	Contains comment update date

- ICommentVote class

No.	Attributes/ Methods	Scope	Description
01	commentId: mongoose.Types.ObjectId	public	Contains original comment if
02	user: string	public	Contains original user id
03	vote: number	public	Contains vote number

- IField class

No.	Attributes/ Methods	Scope	Description
01	englishLabel: string	public	Contains field english label
02	vietnameseLabel: string	public	Contains field vietnamese label
03	englishDescription: string [0..1]	public	Contains field english description
04	vietnameseDescription: string [0..1]	public	Contains field vietnamese description
05	subjects: object[] [0..1]	public	Contains field's subject list
06	image: object [0..1]	public	Contains field image
07	isDeleted: boolean	public	Contains field visibility
08	createdAt: Date	public	Contains field create date
09	updatedAt: Date	public	Contains field update date

- IGoal class

No.	Attributes/ Methods	Scope	Description
01	user_id: string	public	Contains goal author id
02	title: string	public	Contains goal title
03	description: string [0..1]	public	Contains goal description
04	goal_value: number	public	Contains goal final value
05	deadline: Date [0..1]	public	Contains goal deadline

06	current_value: number [0..1]	public	Contains goal current value, when current value equals or passes the final value, the goal is considered finished.
07	end_time: Date	public	Contains goal end time
08	status: goalStatuses	public	Contains goal status

- IInvitation class

No.	Attributes/ Methods	Scope	Description
01	groupId: mongoose.Types.ObjectId	public	Contains invitation original group id
02	senderId: String	public	Contains invitation original sender id
03	receiverId: String	public	Contains invitation original receiver id
04	message: string	public	Contains invitation message
05	expiration: Date	public	Contains invitation expiration date
06	createdAt: Date	public	Contains invitation create date
07	updatedAt: Date	public	Contains invitation update date

- IGroup class

No.	Attributes/ Methods	Scope	Description
01	name: string	public	Contains group name
02	leaderId: string	public	Contains group leader id
03	description: string	public	Contains group description

04	isPublic: boolean	public	Contains group public visibility
05	isActive: boolean	public	Contains group visibility
06	canInvite: boolean	public	Contains group rule whether members except leader can send invitation
07	canAddQuestionSet: boolean	public	Contains group rule whether members except leader can share question set
08	createdAt: Date	public	Contains group create date
09	updatedAt: Date	public	Contains group update date

- IGroupMember class

No.	Attributes/ Methods	Scope	Description
01	groupId: mongoose.Types.ObjectId	public	Contains group member original group id
02	userId: string	public	Contains group member id
03	isBan: boolean	public	Contains group member ban status
04	createdAt: Date	public	Contains group member join date
05	updatedAt: Date	public	Contains group member update date

- IGroupQuestionSet class

No.	Attributes/ Methods	Scope	Description
01	groupId: mongoose.Types.ObjectId	public	Contains group question set original group id
02	questionSetId: mongoose.Types.ObjectId	public	Contains group question set id
03	userId: string	public	Contains group member id that share the question set

04	createdAt: Date	public	Contains group question set join date
05	updatedAt: Date	public	Contains group question set update date

- IQuestion class

No.	Attributes/ Methods	Scope	Description
01	questionTerm: string	public	Contains question term
02	questionOptions: [questionOption]	public	Contains question options as id list
03	questionSetId: mongoose.Types.ObjectId	public	Contains question original question set id
04	order: number	public	Contains question order in a question set
05	type: QuestionType	public	Contains question type
06	image: object	public	Contains question image
07	createdAt: Date	public	Contains question create date
08	updatedAt: Date	public	Contains question update date
09	totalOption: number	public	Contains question total option
10	totalCorrectOption: number	public	Contains question total correct option

- IQuestionProgress class

No.	Attributes/ Methods	Scope	Description
01	questionSetProgressId: mongoose.Types.ObjectId	public	Contains question progress id
02	question: mongoose.Types.ObjectId	public	Contains question progress original question id

03	isMarked: boolean	public	Contains question progress mark status
04	type: QuestionProgressStatus	public	Contains question progress type
05	createdAt: Date	public	Contains question progress create date
06	updatedAt: Date	public	Contains question progress update date

- IQuestionSet class

No.	Attributes/ Methods	Scope	Description
01	title: string	public	Contains question set title
02	description: string	public	Contains question set description
03	author: string	public	Contains question set author
04	subjects: ObjectId[]	public	Contains question set subject list
05	questions: object[]	public	Contains question set question list
06	isPublic: boolean	public	Contains question set public visibility
07	isActive: boolean	public	Contains question set visibility
08	isDeleted: boolean	public	Contains question set delete status
09	totalQuestions: number	public	Contains question set total question
10	totalRatingSum: number	public	Contains question set rating sum
11	ratingCount: number	public	Contains question set rating count

12	averageRating: number	public	Contains question set average rating
13	createdAt: Date	public	Contains question set create date
14	updatedAt: Date	public	Contains question set update date

- IQuestionSetProgress class

No.	Attributes/ Methods	Scope	Description
01	userId: string	public	Contain question set progress original user id
02	questionSetId: ObjectId	public	Contain question set progress original question set id
03	isFavorite: boolean	public	Contain question set progress favorite status
04	rating: number	public	Contain question set progress rating of each user
05	lastAccessed: Date	public	Contain question set last access time of a user
06	createdAt: Date	public	Contain question set progress create date
07	updatedAt: Date	public	Contain question set progress update date

- IQuizResult class

No.	Attributes/ Methods	Scope	Description
01	userId: string	public	Contains quiz result of original user id
02	questionSetId: ObjectId	public	Contains quiz result of original question set id
03	timeSpent: string	public	Contains quiz result time spent of a user

04	totalScore: number	public	Contains quiz result total score
05	createdAt: Date	public	Contains quiz result create date
06	updatedAt: Date	public	Contains quiz result update date

- IReport class

No.	Attributes/ Methods	Scope	Description
01	user_id: string	public	Contains report original reported id
02	reportedId: ObjectId	public	Contains report original entity id
03	type: reportType	public	Contains report type
04	title: string	public	Contains report title
05	description: string	public	Contains report description
06	isResolved: boolean	public	Contains report resolve status
07	createdAt: Date	public	Contains report create date
08	updatedAt: Date	public	Contains report update date

- IStatistics class

No.	Attributes/ Methods	Scope	Description
01	date: Date	public	Contains generated date of statistics
02	totalUsers: number	public	Contains total users of current date

03	totalQuestionSets: number	public	Contains total question sets of current date
04	totalGroups: number	public	Contains total groups of current date
05	totalReports: number	public	Contains total reports of current date
06	deletedUsers: number	public	Contains total deleted users of current date
07	deletedQuestionSets: number	public	Contains total deleted question sets of current date
08	deletedGroups: number	public	Contains total deleted groups of current date
09	deletedReports: number	public	Contains total deleted reports of current date
10	solvedReports: number	public	Contains total solved reports of current date

- ISubject class

No.	Attributes/ Methods	Scope	Description
01	englishLabel: string	public	Contains subject english label
02	vietnameseLabel: string	public	Contains subject vietnamese label
03	englishDescription: string [0..1]	public	Contains subject english description
04	vietnameseDescription: string [0..1]	public	Contains subject vietnamese description
05	fieldId: mongoose.Types.ObjectId	public	Contains subject original field id
06	image: object [0..1]	public	Contains subject image
07	isDeleted: boolean	public	Contains subject visibility

08	createdAt: Date	public	Contains subject create date
09	updatedAt: Date	public	Contains subject update date

- IUser class

No.	Attributes/ Methods	Scope	Description
01	clerkId: string	public	Contains user id generated by Clerk
02	username: string	public	Contains user username
03	email: string	public	Contains user email
04	firstName: string	public	Contains user first name
05	lastName: string	public	Contains user last name
06	fullName: string [0..1]	public	Contains user full name
07	imageUrl: string	public	Contains user image url
08	bio: string	public	Contains user bio
09	neuron: number	public	Contains user neuron number
10	theme: SettingThemes	public	Contains user preferred theme
11	language: SettingLanguages	public	Contains user preferred language
12	restTime: number	public	Contains user custom rest time
13	interests: mongoose.Types.ObjectId[]	public	Contains user interest list of subject

14	status: string	public	Contains user status
15	showQuote: boolean	public	Contains user preferred visibility of quotes
16	showGoalWarning: boolean	public	Contains user preferred visibility of due goals
17	showEmail: boolean	public	Contains user preferred visibility of email
18	showFullName: boolean	public	Contains user preferred visibility of full name
19	isActive: boolean	public	Contains user visibility
20	isDeleted: boolean	public	Contains user deleted status
21	createdAt: Date	public	Contains user sign up date
22	updatedAt: Date	public	Contains user update date

- IUserAchievement class

No.	Attributes/ Methods	Scope	Description
01	userId: string	public	Contains original user id that got the achievement
02	achievement: mongoose.Types.ObjectId	public	Contains achievement id
03	progress: number	public	Contains progress of current achievement
04	completed: boolean	public	Contains achievement complete status
05	completedAt: Date [0..1]	public	Contains achievement complete time
06	createdAt: Date	public	Contains achievement create date

07	updatedAt: Date	public	Contains achievement update date
----	-----------------	--------	----------------------------------

- IUserFollow class

No.	Attributes/ Methods	Scope	Description
01	followerId: string	public	Contains follower id
02	followeeId: string	public	Contains followee id
03	createdAt: Date	public	Contains follow status create date
04	updatedAt: Date	public	Contains follow status update date

- IQuizResultQuestion class

No.	Attributes/ Methods	Scope	Description
01	quizResultId: mongoose.Types.ObjectId	public	Contains original quiz result id
02	questionId: mongoose.Types.ObjectId	public	Contains original question id
03	questionTerm: string	public	Contains question term
04	questionOptions: [object]	public	Contains list of question options
05	userAnswer: UserAnswer	public	Contains user answer

4.5 Middleware Classes

- AuthMiddleware class

No.	Attributes/ Methods	Scope	Description
01	isAuthorized(req: Request, res: Response,	public	Handles user authorization before sending the request data to the next middleware or controller

	next: NextFunction): Promise<Response undefined>		
02	hasPermission(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles Administrator authorization before sending the request data to the next middleware or controller

- ValidationMiddleware class

No.	Attributes/ Methods	Scope	Description
01	validateRequestData(validationSchema: z.ZodObject<any, any>): (req: Request, res: Response, next: NextFunction) => void	public	Handles validating request data before sending it to the next middleware or controller

- UploadMiddleware class

No.	Attributes/ Methods	Scope	Description
01	storage: multer:StorageEngine	private	Handle configure storage
02	fileFilter(req: Request, file: Express.Multer.File, cb: multer.FileFilterCallback) => void	private	Contain method to configure file filter
03	multer: multer.Multer	public	Handle creating the multer instance

- ErrorMiddleware class

No.	Attributes/ Methods	Scope	Description
01	errorHandler(err: Error, req: Request, res: Response,	public	Handles error thrown by previous middlewares and returns as Response data

	next: NextFunction): Promise<Response undefined>		
--	--------------------------------------------------------	--	--

4.6 Controller Classes

- CommentController class

No.	Attributes/ Methods	Scope	Description
01	getAllQuestionsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve all comments
02	getCommentByIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve a comment by ID
03	getCommentRepliesHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve comment replies
04	createCommentHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to create a new comment
05	updateCommentHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to update an existing comment
06	deleteCommentHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to delete an existing comment
07	voteCommentHandler(req: Request, res: Response,	public	Handles the request to vote on a comment

	next: NextFunction): Promise<Response undefined>		
08	getUserVoteHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve a user's vote on a comment
09	acceptCommentHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to mark a comment as accepted

- FieldController class

No.	Attributes/ Methods	Scope	Description
01	getAllFieldsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve all fields
02	getFieldByIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve a field
03	createFieldHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to create a new field
04	updateFieldHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to update an existing field
05	deleteFieldHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to delete an existing field

- GoalController class

No.	Attributes/ Methods	Scope	Description
01	getGoalByUserIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve all goals by specific user id
02	getGoalByIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve a goal
03	createGoalHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to create a new goal
04	updateGoalHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to update an existing goal
05	deleteGoalHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to delete an existing goal

- GroupController class

No.	Attributes/ Methods	Scope	Description
01	getAllGroupsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve all groups
02	getAllGroupMembersHandler(req: Request,	public	Handles the request to retrieve all group members

	res: Response, next: NextFunction): Promise<Response undefined>		
03	getAllMembersByGroupIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve all members from specific group
04	getAllGroupQuestionSetsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve all group question sets
05	getAllQuestionSetsByGroupIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve all questions from specific group
06	getGroupByIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve a group
07	createGroupHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to create a new group
08	createGroupMemberHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to create a new group member
09	createGroupQuestionSetHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to create a new group question set

10	<pre>updateGroupByIdHandler(req: Request, res: Response, next: NextFunction) : Promise<Response undefined></pre>	public	Handles the request to update a group by ID
11	<pre>deleteGroupMemberByIdHandler(req: Request, res: Response, next: NextFunction) : Promise<Response undefined></pre>	public	Handles the request to delete a group member by ID
12	<pre>deleteGroupQuestionSetByIdHandler(req: Request, res: Response, next: NextFunction) : Promise<Response undefined></pre>	public	Handles the request to delete a group question set by ID
13	<pre>createInvitationHandler(req: Request, res: Response, next: NextFunction) : Promise<Response undefined></pre>	public	Handles the request to create a new invitation
14	<pre>deleteGroupQuestionSetByIdHandler(req: Request, res: Response, next: NextFunction) : Promise<Response undefined></pre>	public	Handles the request to retrieve all invitations
15	<pre>getInvitationByIdHandler(req: Request, res: Response, next: NextFunction) : Promise<Response undefined></pre>	public	Handles the request to retrieve an invitation by ID
16	<pre>deleteGroupAllHandler(req: Request,</pre>	public	Handles the request to delete a group and its data

	<pre>res: Response, next: NextFunction) : Promise<Response undefined></pre>		
17	<pre>isGroupMemberBannedHandler(req: Request, res: Response, next: NextFunction) : Promise<Response undefined></pre>	public	Middleware to check if a group member is banned
18	<pre>invitationProcessHandler(req: Request, res: Response, next: NextFunction) : Promise<Response undefined></pre>	public	Handles the request to accept or decline an invitation
19	<pre>banGroupMemberHandler(req: Request, res: Response, next: NextFunction) : Promise<Response undefined></pre>	public	Middleware to check if a group member is banned
20	<pre>ordinateGroupLeaderHandler(req: Request, res: Response, next: NextFunction) : Promise<Response undefined></pre>	public	Handles the request to accept or decline an invitation
21	<pre>banGroupHandler(req: Request, res: Response, next: NextFunction) : Promise<Response undefined></pre>	public	Handles the request to ban or unban a group member
22	<pre>getAllGroupByUserIdHandler(req: Request, res: Response, next: NextFunction) : Promise<Response undefined></pre>	public	Handles the request to retrieve groups by user ID

- ProgressController class

No.	Attributes/ Methods	Scope	Description
01	initializeUserProgressHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to initialize user progress for a question set
02	getUserQuestionSetProgressHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve user progress for a question set
03	getUserQuestionProgressHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve user progress for questions in a set
04	getUserAllQuestionSetProgressesHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve all user question set progresses
05	updateQuestionSetProgressHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to update question set progress
06	updateQuestionsProgressHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to update questions progress in a set
07	updateQuestionProgressHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to update a specific question progress

- QuestionController class

No.	Attributes/ Methods	Scope	Description
01	getAllQuestionsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve all questions
02	getQuestionByIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve a question
03	createQuestionHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to create a new question
04	updateQuestionHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to update an existing question
05	deleteQuestionHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to delete an existing question

- QuestionSetController class

No.	Attributes/ Methods	Scope	Description
01	getAllQuestionSetsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve all question sets
02	getQuestionSetByIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve a question set

03	<code>createQuestionSetHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to create a new question set
04	<code>updateQuestionSetHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to update an existing question set
05	<code>deleteQuestionSetHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to delete an existing question set
06	<code>blockQuestionSetHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to block a question set
07	<code>unblockQuestionSetHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to unblock a question set
08	<code>getUserInterestQuestionSetsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to retrieve question sets based on user interests
09	<code>exportQuestionSetAsDocHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to export a question set as a DOC file
10	<code>exportSampleQuestionSetAsDocHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to export a sample question set as a DOC file

- QuizController class

No.	Attributes/ Methods	Scope	Description
01	getQuizHistoryHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve quiz history of a learner
02	getQuizDetailHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to a quiz detail of a learner
03	createQuizHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to create a new quiz for user
04	submitQuizHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to calculates quiz attempt of a user

- ReportController class

No.	Attributes/ Methods	Scope	Description
01	getAllReportsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve all reports
02	getReportByIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve a report
03	createReportHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to create a new report

04	<code>updateReportHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to update an existing report
05	<code>deleteReportHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to delete an existing report
06	<code>getReportsByReportUserIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to retrieve reports reported to a user id

- SubjectController class

No.	Attributes/ Methods	Scope	Description
01	<code>getAllSubjectsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to retrieve all subjects
02	<code>getSubjectByIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to retrieve a subject
03	<code>createSubjectHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to create a new subject
04	<code>updateSubjectHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to update an existing subject
05	<code>deleteSubjectHandler(req: Request,</code>	public	Handles the request to delete an existing subject

	<pre>res: Response, next: NextFunction): Promise<Response undefined></pre>		
--	-------------------------------------------------------------------------------------	--	--

- UserController class

No.	Attributes/ Methods	Scope	Description
01	<pre>getAllClerkUsersHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></pre>	public	Handles the request to retrieve users from Clerk
02	<pre>getOneClerkUserHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></pre>	public	Handles the request to retrieve one user from Clerk
03	<pre>getAllUsersHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></pre>	public	Handles the request to retrieve all users
04	<pre>getUserByIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></pre>	public	Handles the request to retrieve one user
05	<pre>syncClerkUserHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></pre>	public	Handles the request to sync user between the server and Clerk
06	<pre>banUserHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></pre>	public	Handles the request to block a user
07	<pre>unbanUserHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></pre>	public	Handles the request to unblock a user

08	<code>getCurrentUserHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to unban a user
09	<code>updateUserHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to update a user

- UserFollowController class

No.	Attributes/ Methods	Scope	Description
01	<code>followUserHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to follow new learner
02	<code>unfollowUserHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to unfollow a learner
03	<code>getAllFollowersHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to retrieve all followers of an account
04	<code>getAllFolloweesHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to retrieve all followings of an account
05	<code>checkFollowStatusHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to check follower and following status of an account

- StatisticController class

No.	Attributes/ Methods	Scope	Description
01	getAllStatisticsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve all statistics with filters
02	getShowLast7daysHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve statistics for the last 7 days
03	getShowLast30daysHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve statistics for the last 30 days
04	getShowLastYearHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve statistics for the last year
05	getTotalUserCountHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve the total user count
06	getTotalQuestionSetCountHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve the total question set count
07	getTotalGroupHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined>	public	Handles the request to retrieve the total group count
08	getTotalReportHandler(req: Request, res: Response, next: NextFunction	public	Handles the request to retrieve the total report count

	<code>): Promise<Response undefined></code>		
09	<code>getQuestionSetBySubjectHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to retrieve question sets by subject
10	<code>getTopUsersHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to retrieve top users
11	<code>getTopGroupsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to retrieve top groups

- AchievementController class

No.	Attributes / Methods	Scope	Description
01	<code>getAllAchievementsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to retrieve all achievements
02	<code>getAchievementByIdHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to retrieve an achievement by ID
03	<code>getUserAchievementsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></code>	public	Handles the request to retrieve a user's achievements with progress

04	<pre>getCurrentUserAchievementsHandler(req: Request, res: Response, next: NextFunction): Promise<Response undefined></pre>	public	Handles the request to retrieve the current user's achievements
----	------------------------------------------------------------------------------------------------------------------------------------------------------	--------	-----------------------------------------------------------------

4.7 Service Classes

- CommentService class

No.	Attributes/ Methods	Scope	Description
01	<pre>getAllComments(options?: any): Promise<IComment[]></pre>	public	Handles the request to retrieves all comments with optional filters
02	<pre>getCommentById(options?: any): Promise<IComment></pre>	private	Retrieves a comment by ID with optional filters
03	<pre>getCommentReplies(id: string): Promise<IComment[]></pre>	private	Handles the request to retrieves all replies for a specific comment
04	<pre>createNewComment(data: any): Promise<IComment></pre>	public	Handles the request to creates a new comment with validation
05	<pre>updateComment(id: string, questionSetId: string, userId: string, data: CommentData): Promise<IComment></pre>	public	Handles the request to updates an existing comment with user validation
06	<pre>deleteComment(id: string, questionSetId: string, userId: string): Promise<IComment></pre>	public	Handles the request to deletes a comment with user validation
07	<pre>voteComment(commentId: string, userId: string, voteType: Vote): Promise<{ comment: IComment null; vote: ICommentVote null ></pre>	public	Handles the request to handles voting on a comment

08	<code>getUserVoteOnComment(commentId: string, userId: string): Promise<ICommentVote null></code>	public	Handles the request to retrieves a user's vote on a specific comment
09	<code>acceptComment(commentId: string, questionSetId: string, userId: string): Promise<IComment></code>	public	Handles the request to marks a comment as accepted by the question set creator
10	<code>getComment(commentId: string): Promise<IComment></code>	public	Handles the request to retrieves a comment by its ID

- FieldService class

No.	Attributes/ Methods	Scope	Description
01	<code>getAllFields(): Promise<IField[]></code>	public	Handles the request to retrieve all fields
02	<code>getFieldById(id: string): Promise<IField></code>	public	Handles the request to retrieve a field
03	<code>createNewField(fieldData: any): Promise<IField></code>	public	Handles the request to create a new field
04	<code>updateFieldById(id: string, fieldData: any): Promise<IField></code>	public	Handles the request to update an existing field
05	<code>deleteFieldById(id: string): Promise<IField></code>	public	Handles the request to delete an existing field

- GoalService class

No.	Attributes/ Methods	Scope	Description
01	<code>getGoalByUserId(userId: string): Promise<IGoal[]></code>	public	Handles the request to retrieve all goals of a learner

02	<code>getGoalById(id: string)</code> : Promise<IGoal null>	public	Handles the request to retrieve a goal of a learner
03	<code>createNewGoal(data: GoalData)</code> : Promise<IGoal null>	public	Handles the request to create a new goal
04	<code>updateGoal(id: string, data: GoalData)</code> : Promise<IGoal null>	public	Handles the request to update an existing goal
05	<code>deleteGoal(id: string)</code> : Promise<IGoal null>	public	Handles the request to delete an existing goal

- GroupService class

No.	Attributes / Methods	Scope	Description
01	<code>getAllGroups(option: PaginationOptions)</code> : Promise<IGroup[]>	public	Handles the request to retrieve all groups
02	<code>getAllGroupMembers(option: PaginationOptions)</code> : Promise<IGroupMember[]>	public	Handles the request to retrieve all group members
03	<code>getAllGroupQuestionSets(option: PaginationOptions)</code> : Promise<IGroupQuestionSet[]>	public	Handles the request to retrieve all group question sets
04	<code>getGroupById(id: string)</code> : Promise<IGroup null>	public	Handles the request to retrieve a group by ID
05	<code>getGroupMemberById(id: string)</code> : Promise<IGroupMember null>	public	Handles the request to retrieve a group member by ID

06	<code>getMemberfromGroupId(groupId: string, memberId: string): Promise<IGroupMember[]></code>	public	Handles the request to retrieve a member from a group
07	<code>isGroupMemberAvailable(groupId: string, userId: string): Promise<boolean></code>	public	Handles the request to check if a user is a member of a group
08	<code>isGroupQuestionSetAvailable(groupId: string, questionSetId: string): Promise<boolean></code>	public	Handles the request to check if a question set is available in a group
09	<code>createNewGroup(data: GroupRequestData): Promise<IGroup null></code>	public	Handles the request to create a new group
10	<code>createNewGroupMember(data: GroupMemberRequestData): Promise<IGroupMember null></code>	public	Handles the request to create a new group member
11	<code>createNewGroupQuestionSet(data: GroupQuestionSetRequestData): Promise<IGroupQuestionSet null></code>	public	Handles the request to create a new group question set
12	<code>updateGroupById(id: string, data: GroupRequestData): Promise<IGroup null></code>	public	Handles the request to update a group by ID
13	<code>updateGroupMemberById(id: string, data: GroupMemberRequestData): Promise<IGroupMember null></code>	public	Handles the request to update a group member by ID
14	<code>deleteGroupMemberById(id: string): Promise<IGroupMember null></code>	public	Handles the request to delete a group member by ID

15	<code>deleteGroupQuestionSetById(id: string, memberId: string): Promise<IGroupQuestionSet null></code>	public	Handles the request to delete a group question set by ID
16	<code>createNewInvitation(data: InvitationrequestData): Promise<IInvitation null></code>	public	Handles the request to create a new invitation
17	<code>getInvitationById(id: string): Promise<IInvitation null></code>	public	Handles the request to retrieve an invitation by ID
18	<code>getAllInvitations(option: Paginationoptions): Promise<IInvitation[]></code>	public	Handles the request to retrieve all invitations
19	<code>getAllMembersByGroupId(groupId: string, option: Paginationoptions): Promise<IGroupMember[]></code>	public	Handles the request to retrieve all members of a group by group ID
20	<code>deleteGroupAll(id: string, userId: string): Promise<IGroup null></code>	public	Handles the request to delete a group and all associated data
21	<code>isGroupMemberBanned(groupId: string, userId: string): Promise<boolean></code>	public	Handles the request to check if a group member is banned
22	<code>isGroupAddQuestionSetAllowed(groupId: string): Promise<boolean></code>	public	Handles the request to check if adding a question set is allowed in a group
23	<code>isGroupInviteAllowed(groupId: string): Promise<boolean></code>	public	Handles the request to check if inviting members is allowed in a group

24	<code>deleteInvitationById(id: string): Promise<IInvitation null></code>	public	Handles the request to delete an invitation by ID
25	<code>ordinateGroupLeader(groupId: string, newLeaderId: string, currentLeaderId: string): Promise<IGroup null></code>	public	Handles the request to ordinate a new group leader
26	<code>createGroupByLeader(data: GroupRequestData, userId: string): Promise<IGroup null></code>	public	Handles the request to create a group by a leader
27	<code>createGroupQuestionSetById(data: GroupQuestionSetrequestData, userId: string): Promise<IGroupQuestionSet null></code>	public	Handles the request to create a group question set by ID
28	<code>createGroupMemberById(data: GroupMemberrequestData): Promise<IGroupMember null></code>	public	Handles the request to create a group member by ID
29	<code>invitationProcess(id: string, status: string, userId: string): Promise<IInvitation null></code>	public	Handles the request to process an invitation (accept/decline)
30	<code>getNonGroupMemberId(groupId: string, senderId: string, receiverId: string): Promise<IGroupMember null></code>	private	Handles the request to determine the non-group member ID for invitation processing
31	<code>getAllGroupByUserId(userId: string, option: PaginationOptions): Promise<IGroup[]></code>	public	Handles the request to retrieve all groups for a user by user ID

32	<code>createInvitationByUserId(data: InvitationrequestData, userId: string): Promise<IInvitation null></code>	public	Handles the request to create an invitation by user ID
33	<code>banGroupId(groupId: string, status: string): Promise<IGroup null></code>	public	Handles the request to ban or unban a group
34	<code>banGroupMember(groupId: string, memberId: string, status: string): Promise<IGroupMember null></code>	public	Handles the request to ban or unban a group member
35	<code>getAllQuestionSetByGroupId(groupId: string, option: PaginationOptions): Promise<IGroupQuestionSet []></code>	public	Handles the request to retrieve all question sets for a group by group ID

- ProgressService class

No.	Attributes / Methods	Scope	Description
01	<code>getUserProgressForQuestionSet(questionsetId: string, userId: string): Promise<IQuestionSetProgress null></code>	public	Handles the request to retrieve user progress for a specific question set
02	<code>createUserProgress(userId: string, questionsetId: string): Promise<{ questionSetProgress: IQuestionSetProgress; questionProgress: IQuestionProgress[] }></code>	public	Handles the request to create or retrieve user progress for a question set
03	<code>updateQuestionSetProgressById(questionsetId: string, userId: string, data: Partial): Promise<IQuestionSetProgress null></code>	public	Handles the request to update question set progress by ID

04	<code>updateRating(questionSet: IQuestionSet, newRating: number, oldRating: number): Promise<IQuestionProgress null></code>	private	Handles the request to update the aggregate rating of a question set
05	<code>getQuestionsProgressBelongToUser(userId: string, questionSetId: string): Promise<IQuestionProgress[]></code>	public	Handles the request to retrieve all question progress records for a user in a question set
06	<code>updateQuestionsProgress(userId: string, questionSetId: string, data: any): Promise<IQuestionProgress[]></code>	public	Handles the request to update multiple question progress records
07	<code>getUserAllQuestionSetProgresses(userId: string): Promise<IQuestionSetProgress[]></code>	public	Handles the request to retrieve all question set progresses for a user
08	<code>updateQuestionProgressById(id: string, data: any): Promise<IQuestionProgress null></code>	public	Handles the request to update a single question progress by ID

- QuestionService class

No.	Attributes/ Methods	Scope	Description
01	<code>getAllQuestions(): Promise<IQuestion[] null></code>	public	Handles the request to retrieve all questions
02	<code>getQuestionById(id: string): Promise<IQuestion null></code>	public	Handles the request to retrieve a question
03	<code>createNewQuestion(data: any): Promise<IQuestion null></code>	public	Handles the request to create a new question
04	<code>updateQuestionById(id: string, data: any): Promise<IQuestion null></code>	public	Handles the request to update an existing question

05	<code>deleteQuestionById(id: string)</code> : Promise<IQuestion null>	public	Handles the request to delete an existing question
----	---------------------------------------------------------------------------------------	--------	----------------------------------------------------

- QuestionSetService class

No.	Attributes / Methods	Scope	Description
01	<code>getAllQuestionSets(queryParams: <QuestionSetFilters>)</code> : Promise<IQuestionSet[]>	public	Handles the request to retrieve all question sets with pagination and filters
02	<code>getQuestionSetById(id: string)</code> : Promise<IQuestionSet null>	public	Handles the request to retrieve a question set by ID
03	<code>createNewQuestionSet(data: QuestionsetData, userId: string)</code> : Promise<IQuestionSet null>	public	Handles the request to create a new question set
04	<code>updateQuestionSetById(id: string, data: any, userId: string)</code> : Promise<IQuestionSet null>	public	Handles the request to update a question set by ID
05	<code>deleteQuestionSetById(id: string)</code> : Promise<IQuestionSet null>	public	Handles the request to delete a question set by ID
06	<code>isQuestionSetCreator(questionsetId: string, userId: string)</code> : Promise<IQuestionSet null>	public	Handles the request to check if a user is the creator of a question set
07	<code>blockQuestionSet(questionsetId: string)</code> : Promise<IQuestionSet null>	public	Handles the request to block a question set

08	<code>unblockQuestionSet(questionSetId: string)</code> : Promise<IQuestionSet null>	public	Handles the request to unblock a question set
09	<code>getUserInterestQuestionSets(userId: string, queryParams: QueryParams)</code> : Promise<IQuestionSet null>	public	Handles the request to retrieve question sets matching a user's interests
10	<code>getUserInterestsById(userId: string)</code> : Promise<string[]>	private	Handles the request to retrieve a user's interests by ID
11	<code>exportQuestionSetAsDoc(id: string)</code> : Promise<{ buffer: ArrayBufferLike; title: string }>	public	Handles the request to export a question set as a document
12	<code>exportSampleQuestionSetAsDoc(): Promise<IQuestionSet null></code>	public	Handles the request to export a sample question set as a document

- QuizService class

No.	Attributes/ Methods	Scope	Description
01	<code>createNewQuiz(data: QuizCreationPayload)</code> : Promise<Quiz>	public	Handles the request to create a new quiz
02	<code>submitQuiz(data: QuizSubmissionPayload)</code> : Promise<QuizSubmissionResult>	public	Handles the request to submit the finished quiz
03	<code>getQuizHistory(query?: any)</code> : Promise<IQuizResult[]>	public	Handles the request to retrieve quiz history of a learner
04	<code>getQuizResultDetail(id: string)</code> : Promise<IQuizResult>	public	Handles the request to retrieve quiz attempt detail of a learner

- ReportService class

No.	Attributes/ Methods	Scope	Description
01	getAllReports(options: PaginationOptions): Promise<IReport[] PaginatedResponse<IReport>>	public	Handles the request to retrieve all reports
02	getReportById(id: string): Promise<IReport null>	public	Handles the request to retrieve a report
03	getReportByReportUserId(report_user_id: string): Promise<IReport[]>	public	Handles the request to retrieve a report based on reported user id
04	createNewReport(data: ReportData): Promise<IReport null>	public	Handles the request to create a new question set
05	updateReportById(id: string, data: ReportData): Promise<IReport null>	public	Handles the request to update an existing question set
06	deleteReportById(id: string): Promise<IReport null>	public	Handles the request to delete an existing question set

- SubjectService class

No.	Attributes/ Methods	Scope	Description
01	getAllSubjects(): Promise<ISubject[]>	public	Handles the request to retrieve all subjects
02	getSubjectById(id: string): Promise<ISubject>	public	Handles the request to retrieve a subject
03	createNewSubject(data: SubjectData): Promise<ISubject>	public	Handles the request to create a new subject
04	updateSubjectById(id: string, data: SubjectData): Promise<ISubject>	public	Handles the request to update an existing subject

05	<code>deleteSubjectById(id: string): Promise<ISubject></code>	public	Handles the request to delete an existing subject
----	---------------------------------------------------------------------------------	--------	---------------------------------------------------

- UserService class

No.	Attributes/ Methods	Scope	Description
01	<code>getAllClerkUsers(): Promise<any></code>	public	Handles the request to retrieve all users of Clerk
02	<code>getOneClerkUser(clerkUserId: string): Promise<any></code>	public	Handles the request to retrieve a user of Clerk
03	<code>getOneClerkUserInMongo(clerkUserId: string): Promise<IUser null></code>	public	Handles the request to retrieve all users of the server
04	<code>getAllClerkUsersInMongo(queryParams: QueryParams<UserFilters>): Promise<IUser[]> PaginatedResponse<IUser >></code>	public	Handles the request to retrieve a user of the server
05	<code>syncClerkUserToDatabase(headers: any, payload: any): Promise<void></code>	public	Handles the request to sync user data between the server and Clerk
06	<code>blockUserById(id: string): Promise<boolean></code>	public	Handles the request to block a user
07	<code>unblockUserById(id: string): Promise<boolean></code>	public	Handles the request to unblock a user
08	<code>updateUserById(id: string, data: any): Promise<IUser null></code>	public	Handles the request to unblock a user by ID

- UserFollowService class

No.	Attributes/ Methods	Scope	Description
01	async followUser(followerId: string followeeId: string): Promise<IUserFollow>	public	Handles the request to follow a user
02	async unfollowUser(followerId: string followeeId: string): Promise<IUserFollow>	public	Handles the request to unfollow a user
03	getAllFollowers(userId: string): Promise<{ followers: IUserFollow[]; total: number }>	public	Handles the request to retrieve all followers of a user
04	getAllFollowees(userId: string): Promise<{ followees: IUserFollow[]; total: number }>	public	Handles the request to retrieve all followings of a user
05	checkFollowStatus(followerId: string, followeeId: string): Promise<boolean>	public	Handles the request to check for follower and following status of a user

- AchievementService class

No.	Attributes / Methods	Scope	Description
01	trackAchievement(userId: string, requirementType: string, targetEntity: string, incrementValue = 1): Promise<void>	public	Handles the request to update achievement progress for a user for a specific action

02	<code>updateUserAchievementProgress(userId: string, achievementId: string, incrementValue: number): Promise<IUserAchievement></code>	private	Handles the request to update progress for a specific user achievement and checks for completion
03	<code>completeAchievement(userAchievementId: string): Promise<IUserAchievement></code>	private	Handles the request to mark an achievement as completed and process rewards
04	<code>getUserAchievements(userId: string): Promise<IUserAchievement[]></code>	public	Handles the request to retrieve all achievements for a user with progress
05	<code>getAllAchievements(): Promise<IAchievement[]></code>	public	Handles the request to retrieve all achievements
06	<code>getAchievementById(id: string): Promise<IAchievement></code>	public	Handles the request to retrieve an achievement by ID
07	<code>awardNeuronPoints(userId: string, points: number): Promise<IUser></code>	public	Handles the request to award neuron points to a user and track related achievements

- StatisticsService class

No.	Attributes / Methods	Scope	Description
01	<code>getAllStatistics(queryParams: QueryParams<StatisticFilters>): Promise<IStatistic[]></code>	public	Handles the request to retrieve all statistics
02	<code>getShowLast7days(): Promise<IStatistic[]></code>	public	Handles the request to retrieve data from the last 7 days
03	<code>getShowLast30days(): Promise<IStatistic[]></code>	public	Handles the request to retrieve data from the last 30 days
04	<code>getShowLastYear(): Promise<IStatistic[]></code>	public	Handles the request to retrieve data from the last 365 days

05	getTotalUserCount(): Promise<number>	public	Handles the request to retrieve total number of users
06	getTotalQuestionSetCount(): Promise<number>	public	Handles the request to retrieve total number of question sets
07	getTotalGroup(): Promise<number>	public	Handles the request to retrieve total number of groups
08	getTotalReport(): Promise<number>	public	Handles the request to retrieve total number of reports
09	watchDatabaseChanges(): ((() => void)	public	Handle the request to watch all configured collections
10	getTopGroups(): Promise<IStatistic[]>	public	Handles the request to retrieve top 5 group with the most members
11	getTopUsers(): Promise<IStatistic[]>	public	Handles the request to retrieve top 5 user with the most neuron count

4.8 Repository Classes

- CommentRepository class

No.	Attributes / Methods	Scope	Description
01	findAll(options?: any): Promise<IComment[]>	public	Handles the request to retrieve all comments for a question set
02	findById(options?: any): Promise<IComment null>	public	Handles the request to retrieve a comment by ID for a specific question set
03	findRepliesByCommentId(id: string): Promise<IComment[]>	public	Handles the request to retrieve all replies for a specific comment
04	create(data: any): Promise<IComment null>	public	Handles the request to create a new comment

05	<code>update(id: string, data: CommentData): Promise<IComment null></code>	public	Handles the request to update a comment by ID
06	<code>remove(id: string): Promise<IComment null></code>	public	Handles the request to delete a comment and its replies by ID
07	<code>updateVote(commentId: string, userId: string, voteType: Vote): Promise<{ comment: IComment; vote: ICommentVote }></code>	public	Handles the request to update or toggle a user's vote on a comment
08	<code>getUserVote(commentId: string, userId: string): Promise<ICommentVote></code>	public	Handles the request to retrieve a user's vote on a comment
09	<code>unacceptAllComments(filter: any): Promise<void></code>	public	Handles the request to unaccept all comments matching a filter
10	<code>acceptComment(id: string): Promise<IComment></code>	public	Handles the request to mark a comment as accepted

- FieldRepository class

No.	Attributes/ Methods	Scope	Description
01	<code>findAll(): Promise<IField[]></code>	public	Handles the request to retrieve all fields
02	<code>findById(id: string): Promise<IField null></code>	public	Handles the request to retrieve a field
03	<code>create(data: any): Promise<IField null></code>	public	Handles the request to create a new field

04	<code>update(id: string, data: any): Promise<IField null></code>	public	Handles the request to update a created field
05	<code>remove(id: string): Promise<IField null></code>	public	Handles the request to delete a created field

- GoalRepository class

No.	Attributes/ Methods	Scope	Description
01	<code>findById(userId: string): Promise<IGoal[]></code>	public	Handles the request to retrieve goals base on user id
02	<code>findById(id: string): Promise<IGoal null></code>	public	Handles the request to retrieve a goal
03	<code>create(data: GoalData): Promise<IGoal null></code>	public	Handles the request to create a new goal
04	<code>update(id: string, data: Partial<GoalData>): Promise<IGoal null></code>	public	Handles the request to update a created goal
05	<code>remove(id: string): Promise<IGoal null></code>	public	Handles the request to delete a created goal

- ProgressRepository class

No.	Attributes / Methods	Scope	Description
01	<code>createQuestionSetProgress(userId: string, questionSetId: string): Promise<IQuestionSetProgress null></code>	public	Handles the request to create a new question set progress record

02	<code>findByUserIdAndSetId(userId: string, questionSetId: string)</code> : Promise<IQuestionSetProgress null>	public	Handles the request to find a question set progress by user ID and set ID
03	<code>findAllProgressForQuestionSet(questionSetId: string)</code> : Promise<IQuestionSetProgress[]>	public	Handles the request to retrieve all progress records for a question set
04	<code>updateQuestionSetProgress(questionSetId: string, userId: string, data: Partial)</code> : Promise<IQuestionSetProgress null>	public	Handles the request to update a question set progress record
05	<code>findQuestionProgressByQuestionSetProgressId(id: string)</code> : Promise<IQuestionProgress[]>	public	Handles the request to retrieve question progress records by question set progress ID
06	<code>createManyQuestionProgresses(data: any)</code> : Promise<IQuestionProgress[]>	public	Handles the request to create multiple question progress records
07	<code>updateManyQuestionsProgress(updateData: Array<{ _id: string; [key: string]: any }>)</code> : Promise<IQuestionProgress[]>	public	Handles the request to update multiple question progress records
08	<code>getAllQuestionSetProgress(userId: string)</code> : Promise<IQuestionSetProgress[]>	public	Handles the request to retrieve all question set progress records for a user
09	<code>updateQuestionProgress(id: string, data: any)</code> : Promise<IQuestionSetProgress null>	public	Handles the request to update a single question progress record by ID

- GroupRepository class

No.	Attributes / Methods	Scope	Description
01	findAllGroups(options: PaginationOptions)Promise<IGroup[]>	public	Handles the request to retrieve all groups with pagination
02	findGroupById(id: string)Promise<IGroup null>	public	Handles the request to retrieve a group by ID
03	findAllGroupByUserId(userId: string, options: PaginationOptions) Promise<IGroup[]>	public	Handles the request to retrieve all groups for a user with pagination
04	findAllGroupMembers(options: PaginationOptions) Promise<IGroupMember[]>	public	Handles the request to retrieve all group members with pagination
05	findAllGroupQuestionSets(options: PaginationOptions) Promise<IGroupQuestionSet[]>	public	Handles the request to retrieve all group question sets with pagination
06	findAllMembersByGroupId(groupId: string, options: PaginationOptions) Promise<IGroupMember[]>	public	Handles the request to retrieve all members of a group with pagination
07	findGroupMemberById(id: string) Promise<IGroupMember[]>	public	Handles the request to retrieve a group member by ID
08	findMemberfromGroupId(groupId: string, userId: string) Promise<IGroupMember[]>	public	Handles the request to retrieve a member from a group

09	<code>findGroupQuestionSet(groupId: string, questionSetId: string): Promise<IGroupQuestionSet null></code>	public	Handles the request to retrieve a group question set by group ID and question set ID
10	<code>createGroup(data: GroupRequestData): Promise<IGroup null></code>	public	Handles the request to create a new group
11	<code>createGroupMember(data: GroupMemberRequestData): Promise<IGroupMember null></code>	public	Handles the request to create a new group member
12	<code>createGroupQuestionSet(data: GroupQuestionSetRequestData): Promise<IGroupQuestionSet null></code>	public	Handles the request to create a new group question set
13	<code>updateGroup(id: string, data: Partial): Promise<IGroup null></code>	public	Handles the request to update a group by ID
14	<code>updateGroupMember(id: string, data: Partial): Promise<IGroupMember null></code>	public	Handles the request to update a group member by ID
15	<code>removeGroup(id: string): Promise<IGroup null></code>	public	Handles the request to delete a group by ID
16	<code>removeGroupMember(id: string): Promise<IGroupMember null></code>	public	Handles the request to delete a group member by ID
17	<code>removeGroupQuestionSet(id: string): Promise<IGroupQuestionSet null></code>	public	Handles the request to delete a group question set by ID

18	<code>removeGroupAll(groupId: string)</code> : Promise<{ deletedGroup: IGroup }>	public	Handles the request to delete a group and all associated data
19	<code>createInvitation(data: InvitationrequestData)</code> : Promise<IInvitation null>	public	Handles the request to create a new invitation
20	<code>findInvitationById(id: string)</code> : Promise<IInvitation null>	public	Handles the request to retrieve an invitation by ID
21	<code>findAllInvitations(options: PaginationOptions)</code> : Promise<IInvitation[]>	public	Handles the request to retrieve all invitations with pagination
22	<code>isMemberBanned(groupId: string, userId: string)</code> : Promise<boolean>	public	Handles the request to check if a group member is banned
23	<code>isInviteAllowed(groupId: string)</code> : Promise<boolean>	public	Handles the request to check if inviting members is allowed in a group
24	<code>isAddQuestionSetAllowed(groupId: string)</code> : Promise<boolean>	public	Handles the request to check if adding a question set is allowed in a group
25	<code>removeInvitation(id: string)</code> : Promise<IInvitation null>	public	Handles the request to delete an invitation by ID
26	<code>fuzzySearchGroups(options: PaginationOptions)</code> : Promise<IGroup[]>	public	Handles the request to perform a fuzzy search for groups with pagination
27	<code>findAllGroupQuestionSetByGroupId(groupId: string, options: PaginationOptions)</code> : Promise<IGroupQuestionSet[]>	public	Handles the request to retrieve all group question sets for a group with pagination

- QuestionRepository class

No.	Attributes / Methods	Scope	Description
01	findAll(): Promise<IQuestion[]>	public	Handles the request to retrieve all questions
02	findAllByQuestionSetId(id: string): Promise<IQuestion[]>	public	Handles the request to retrieve all questions for a specific question set
03	findById(id: string): Promise<IQuestion null>	public	Handles the request to retrieve a question by ID
04	findByIdAndQuestionSetId(id: string, questionSetId: string): Promise<IQuestion null>	public	Handles the request to retrieve a question by ID and question set ID
05	findAllByQuestionSetId(id: string): Promise<IQuestion[]>	public	Handles the request to retrieve all questions for a specific question set
06	create(data: any): Promise	public	Handles the request to create a new question
07	update(id: string, data: any): Promise<IQuestion null>	public	Handles the request to update a question by ID
08	remove(id: string): Promise<IQuestion null>	public	Handles the request to delete a question by ID

- QuestionSetRepository class

No.	Attributes / Methods	Scope	Description
01	create(data: any, userId: string):Promise<PaginatedResponse<IQuestionSet>>	public	Handles the request to create a new question set with associated questions
02	reconstructQuestionsArray(data: any): any[]	private	Handles the request to reconstruct questions array from flattened form data
03	findAll(queryParams: QueryParams): Promise<IQuestionSet[]>	public	Handles the request to retrieve all question sets with pagination and filters
04	findById(id: string): Promise<IQuestionSet null>	public	Handles the request to retrieve a question set by ID
05	update(id: string, data: any): Promise<IQuestionSet null>	public	Handles the request to update a question set and its associated questions
06	remove(id: string): Promise<IQuestionSet null>	public	Handles the request to soft delete a question set by ID
07	permanentDelete(id: string): Promise<IQuestionSet null>	public	Handles the request to permanently delete a question set by ID
08	search(queryParams: QueryParams):Promise<PaginatedResponse<IQuestionSet>>	public	Handles the request to search for question sets with pagination
09	fuzzySearch(queryParams: QueryParams):Promise<PaginatedResponse<IQuestionSet>>	public	Handles the request to perform a fuzzy search for question sets

10	normalizeQuestionOrder(questionSetId: string, session: mongoose.ClientSession): Promise<IQuestionSet null>	private	Handles the request to normalize the order of questions in a question set
11	findByInterests(interests: string[], queryParams: QueryParams): Promise<IQuestionSet[]>	public	Handles the request to retrieve question sets matching user interests
12	block(questionSetId: string): Promise<IQuestionSet null>	public	Handles the request to block a question set
13	unlock(questionSetId: string): Promise<IQuestionSet null>	public	Handles the request to unlock a question set
14	exportAsDoc(questionsetData: any): Promise<IQuestionSet null>	public	Handles the request to export a question set as a document
15	exportSample(): Promise<IQuestionSet null>	public	Handles the request to export a sample question set as a document

- QuizRepository class

No.	Attributes/ Methods	Scope	Description
01	findAll(query?: any): Promise<IQuizResult[]>	public	Handles the request to retrieve quiz history of a user
02	findById(id: string): Promise<IQuizResult null>	public	Handles the request to retrieve a quiz attempt
03	submit(data: QuizSubmissionData): Promise<IQuizResult null>	public	Handles the request to submit and calculate a quiz attempt

- ReportRepository class

No.	Attributes/ Methods	Scope	Description
01	findAll(options): PaginatedResponse<(typeof reports)[0]>	public	Handles the request to retrieve all reports
02	findById(id: string): Promise<IReport null>	public	Handles the request to retrieve a report
03	findByReportUserId(report_user_id: string): Promise<IReport[]>	public	Handles the request to retrieve reports based on reported id
04	create(reportData: ReportData): Promise<IReport null>	public	Handles the request to create a new report
05	update(id: string, reportData: Partial<ReportData>): Promise<IReport null>	public	Handles the request to update a created report
06	remove(id: string): Promise<IReport null>	public	Handles the request to delete a created report
07	removeMany(ids: string[]): Promise<IReport[]>	public	Handles the request to delete multiple reports by IDs

- SubjectRepository class

No.	Attributes/ Methods	Scope	Description
01	findAll(): Promise<ISubject[]>	public	Handles the request to
02	findById(id: string): Promise<ISubject null>	public	Handles the request to
03	create(data: SubjectData	public	Handles the request to

	<code>): Promise<ISubject null></code>		
04	<code>update(id: string, data: SubjectData)</code> <code>: Promise<ISubject null></code>	public	Handles the request to
05	<code>remove(id: string)</code> <code>: Promise<ISubject null></code>	public	Handles the request to

- UserRepository class

No.	Attributes/ Methods	Scope	Description
01	<code>findAll(queryParams: QueryParams<UserFilters>)</code> <code>: Promise<PaginatedResponse<IUser>></code>	public	Handles the request to retrieve all users with pagination and filters
02	<code>findById(clerkId: string)</code> <code>: Promise<IUser null></code>	public	Handles the request to retrieve a user by Clerk ID
03	<code>update(clerkId: string, userData: any)</code> <code>: Promise<IUser null></code>	public	Handles the request to update a user by Clerk ID
04	<code>create(userData: any)</code> <code>: Promise<IUser null></code>	public	Handles the request to create a new user
05	<code>remove(clerkId: string)</code> <code>: Promise<IUser null></code>	public	Handles the request to anonymize a user by Clerk ID
06	<code>fuzzySearch(queryParams: QueryParams<UserFilter>)</code> <code>:Promise<PaginatedResponse<IUse r>></code>	public	Handles the request to perform a fuzzy search for users
07	<code>block(id: string)</code> <code>: Promise<IUser null></code>	public	Handles the request to block a user by Clerk ID

08	<code>unblock(id: string)</code> : Promise<IUser null>	public	Handles the request to unblock a user by Clerk ID
09	<code>incrementNeuronPoints(userId: string, points: number)</code> : Promise	public	Handles the request to increment a user's neuron points

- UserFollowRepository class

No.	Attributes/ Methods	Scope	Description
01	<code>createFollow(followerId: string, followeeId: string)</code> : Promise<IUserFollow>	public	Handles the request to create a new follow relationship
02	<code>removeFollow(followerId: string, followeeId: string)</code> : Promise	public	Handles the request to remove a follow relationship
03	<code>findAllFollowers(userId: string)</code> : Promise<{ followers: IUserFollow[]; total: number }>	public	Handles the request to retrieve all followers of a user
04	<code>findAllFollowees(userId: string)</code> : Promise<{ followees: IUserFollow[]; total: number }>	public	Handles the request to retrieve all followees of a user
05	<code>checkFollow(followerId: string, followeeId: string)</code> : Promise<boolean>	public	Handles the request to check if a follow relationship exists

- StatisticRepository class

No.	Attributes / Methods	Scope	Description
01	findAll(queryParams: QueryParams): Promise<I statistic[]>	public	Handles the request to retrieve all statistics with filters
02	showLast7days(): Promise<IStatistic[]>	public	Handles the request to retrieve statistics for the last 7 days
03	showLast30days(): Promise<IStatistic[]>	public	Handles the request to retrieve statistics for the last 30 days
04	showLastYear(): Promise<IStatistic[]>	public	Handles the request to retrieve statistics for the last year
05	totalUsers(): Promise<{ activeUsers: number; inactiveUsers: number }>	public	Handles the request to retrieve the total count of active and inactive users
06	fiveHighestNeuronUser(): Promise<IUser[]>	public	Handles the request to retrieve the top 5 users with the highest neuron points
07	fiveCrowdestGroups(): Promise<Array<IGroup & memberCount: number }>>	public	Handles the request to retrieve the top 5 groups with the most members
08	totalQuestionSets(): Promise<{ activeQuestionSets: number; inactiveQuestionSets: number }>	public	Handles the request to retrieve the total count of active and inactive question sets
09	totalGroups(): Promise<{ activeGroups: number; inactiveGroups: number }>	public	Handles the request to retrieve the total count of active and inactive groups
10	totalReports(): Promise<{ resolvedReport: number; unResolvedReport: number }>	public	Handles the request to retrieve the total count of resolved and unresolved reports

11	removeFirstDay(): Promise<IStatistic null>	public	Handles the request to remove the oldest statistic entry
12	createOrUpdateDailyStat(data: Partial): Promise<IStatistic>	public	Handles the request to create or update a daily statistic entry
13	totalQuestionSetsPerSubject(): Promise<any[]>	public	Handles the request to retrieve the total question sets per subject

- AchievementRepository class

No.	Attributes / Methods	Scope	Description
01	findAll(): Promise<IAchievement[]>	public	Handles the request to retrieve all achievements
02	findById(id: string): Promise<IAchievement null>	public	Handles the request to retrieve an achievement by ID
03	findByRequirement(requirementType: string, targetEntity: string): Promise<IAchievement[]>	public	Handles the request to retrieve achievements by requirement type and target entity
04	findUserAchievements(userId: string): Promise<IUserAchievement[]>	public	Handles the request to retrieve all user achievements for a user
05	findUserAchievement(userId: string, achievementId: string): Promise<IAchievement null>	public	Handles the request to retrieve a specific user achievement by user ID and achievement ID
06	updateUserAchievementProgress(userId: string, achievementId: string, incrementValue: number): Promise<IAchievement null>	public	Handles the request to update or create user achievement progress

07	completeAchievement(userAchievementId: string): Promise<IAchievement null>	public	Handles the request to mark a user achievement as completed
----	--------------------------------------------------------------------------------------	--------	-------------------------------------------------------------

4.9 Error Classes

- BaseError class

No.	Attributes/ Methods	Scope	Description
01	statusCode: number	public	Contains status code of the error
02	errors: { message:string; context?: {[key: string]: any} }	public	Contains context of the error
03	logging: boolean	public	Contains logging status of the error
04	constructor(message: string): BaseError	public	Initializes the class when invoke

- BadRequestError class

No.	Attributes/ Methods	Scope	Description
01	_code: number	private	Contains code of the error
02	_logging: boolean	private	Contains logging status
03	_context: {[key: string]: any}	private	Contains error context information
04	_statusCode: number	private	Contains static code of the error
05	constructor(params?: { code?: number, message?: string, logging?: boolean, context?: { [key: string]: any } }): BadRequestError	public	Initializes the class when invoke

06	<pre>getErrors: { message:string; context?: {[key: string]: any} }</pre>	public	Returns error information
07	<code>getStatusCode(): boolean</code>	public	Returns status code
08	<code>getLogging(): boolean</code>	public	Returns logging status

- `NotFoundError` class

No.	Attributes/ Methods	Scope	Description
01	<code>_code: number</code>	private	Contains code of the error
02	<code>_logging: boolean</code>	private	Contains logging status
03	<code>_context: {[key: string]: any}</code>	private	Contains error context information
04	<code>_statusCode: number</code>	private	Contains static code of the error
05	<pre>constructor(params?: { code?: number, message?: string, logging?: boolean, context?: { [key: string]: any } }): NotNotFoundError</pre>	public	Initializes the class when invoke
06	<pre>getErrors: { message:string; context?: {[key: string]: any} }</pre>	public	Returns error information
07	<code>getStatusCode(): boolean</code>	public	Returns status code
08	<code>getLogging(): boolean</code>	public	Returns logging status

- UnauthorizedError class

No.	Attributes/ Methods	Scope	Description
01	_code: number	private	Contains code of the error
02	_logging: boolean	private	Contains logging status
03	_context: {[key: string]: any}	private	Contains error context information
04	_statusCode: number	private	Contains static code of the error
05	constructor(params?: { code?: number, message?: string, logging?: boolean, context?: { [key: string]: any } }): Unauthorized	public	Initializes the class when invoke
06	getErrors: { message:string; context?: {[key: string]: any} }	public	Returns error information
07	getStatusCode(): boolean	public	Returns status code
08	getLogging(): boolean	public	Returns logging status

- ForbiddenError class

No.	Attributes/ Methods	Scope	Description
01	_code: number	private	Contains code of the error
02	_logging: boolean	private	Contains logging status
03	_context: {[key: string]: any}	private	Contains error context information
04	_statusCode: number	private	Contains static code of the error

05	<code>constructor(params?: { code?: number, message?: string, logging?: boolean, context?: { [key: string]: any } }): Unauthorized</code>	public	Initializes the class when invoke
06	<code>getErrors: { message:string; context?: {[key: string]: any} }</code>	public	Returns error information
07	<code>getStatusCode(): boolean</code>	public	Returns status code
08	<code>getLogging(): boolean</code>	public	Returns logging status

- DatabaseError class

No.	Attributes/ Methods	Scope	Description
01	<code>constructor(error: any): DatabaseError</code>	public	Initializes the class when invoke
02	<code>processError(error: any): never</code>	private	Determines the type of error returns by the database

V. Software Testing Documentation

1. Scope of Testing

The test plan will have to cover the majority of facets of each test subject, this ensures a high level of reliability, quality and effectiveness. Below is a list of detailed breakdowns for each test aspects:

- **Component and Integration testing:** This is essential for the first phase of testing, as they focus on verifying the correctness of individual components of the system. Unit and integration testing is commonly used here as they help catch bugs early on during the development process, making them easier to fix later.
- **System and User-Centric testing:** These test types shift their focus on the fully integrated system, where the overall quality is evaluated and measured on well it meets the user requirement. System and user acceptance testing is chosen to simulate how the system will behave in the real-world scenario.
- **Quality testing:** By focusing on the non-functional requirements aspects of the system, this guarantees that the system not only performs as expected but also secure, reliable and effective, providing a stable yet accessible experience for users. Tests that focus on such criteria's like performance, security, accessibility... will be used to serve such purposes.

The test plan will be carried out in 4 phases: Unit testing, Integration testing, System testing and User Acceptance testing. Each detail will be listed inside the table below

Phase	In-charge	When	Focus	Acceptance criteria
Unit testing	Team members	Perform at the same time with development process, after each component is completed	Verifying each component of code to ensure it operates correctly when tested in small units.	All unit tests are performed successfully without any issue, and the code is compiled following the best practice for relevant technology.
Integration testing	Team members	Perform after components has been tested and prior to over system testing	Verifying the integration components work according to how the requirement is indicated.	All integration attempts are carried out successfully, and the data exchange works as intended.
System testing	Team members	Perform after integration process, where the communication between client and server has been tested	Evaluating the system to ensure it fulfills all the functional and performance requirements.	The system passes all the tests and meets the actual standards and requirements.
User acceptance	Team members	Perform lastly, before launching, allow the	Testing to assess how actual users interact	The system performs well without any major

testing		system to asset the real scenarios	with the system for evaluating usability and meeting business objectives.	issues, and the outcome meets the minimum expectation of actual users.
---------	--	------------------------------------	---------------------------------------------------------------------------	------------------------------------------------------------------------

2. Test Strategy

Following the Scrum methodology, the team decided to make use of Agile testing, which at its core, facilitates the relationship between continuous development and feedback, ensuring the team to quickly and efficiently identify any upcoming problems or issues. This not only strengthens the collaboration between members but also saves time and efforts of identifying and fixing small issues later.

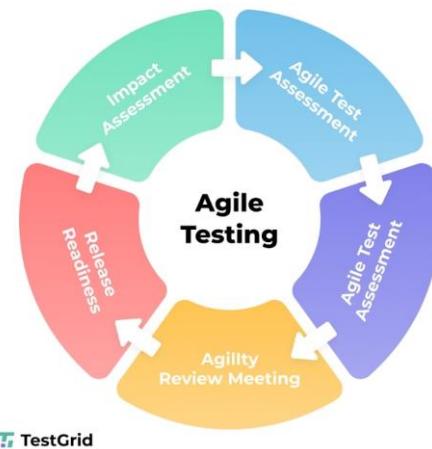


Figure 2: Agile testing²⁴⁹

2.1 Testing Types

Phase	Objective	Technique	Completion Criteria
Unit testing	To verify that the functionality of each unit inside client and server works as intended when used independently.	Utilizes white box technique to test for different types of data and events, and carry them through continuous iterations to ensure correct flow.	All unit tests pass successfully, achieving high coverage and contributing to the ongoing integration and continuous delivery process.
Integration testing	To verify that the communication between client and server work as intended when handling request or response	Utilizes white box technique along with exploratory techniques that enable both client and server team to have a growing structural	The test is considered complete when all critical interactions are functioning as expected and any identified defects have been addressed.

²⁴⁹ <https://testgrid.io/blog/agile-testing/>

		understanding of both sides. While carry out different input type to see how well both side interact with each other	Continuous feedback and iterative refinement may be required to ensure ongoing integration success.
System testing	To verify that the system performs well from start to finish, and still meet the initial requirement and means of interaction.	For system testing, the team focuses on blackbox testing, where other members will try to simulate what they expect the system to work, helps verify that the system is built correctly.	The system meets all defined functional and non-functional criteria, has been tested in a simulated production environment, and shows consistent performance with feedback from users and stakeholders integrated into the final product.
User acceptance testing	To validate that the system is capable of performing well even outside the real world, ensure that the product is ready for the end users.	For UAT, it is preferred to have actual users to execute test cases that are based on real-world scenarios. This type of testing often happens at the end of each milestone of the project. However, occasionally, monkey testing is also used by team members to simulate uncommon activity.	The test is considered complete when users confirm that the software meets their requirements and performs effectively in their operational environment. Feedback and any issues identified are addressed promptly to ensure the software is ready for deployment.

2.2 Test Levels

Test type	Test level			
	Unit	Integration	System	Acceptance
Performance test	X	X	X	X
Accessibility test			X	X
Security test	X	X	X	X
Data integrity test	X	X	X	X

2.3 Supporting Tools

Phase	Tool	Vendor/In-house	Version
Unit test	Manual	Manual	2.0.0
Integration test	Manual	Manual	2.0.0
System test	Manual	Manual	2.0.0
User acceptance test	Manual	Manual	2.0.1

3. Test Plan

3.1 Human Resources

Members	Role	Responsibilities
Huỳnh Nguyễn Gia Bảo	Leader	<ul style="list-style-type: none"> - Manage the overall testing process. - Design test strategies - Create test case for server - Perform test case - Record test result
Mai Quang Vinh	Tester	<ul style="list-style-type: none"> - Create test case for server - Perform test case - Record test result
Mai Nhật Đăng	Tester	<ul style="list-style-type: none"> - Create test case for client - Perform test case - Record test result
Trần Lê Hoàng Long	Tester	<ul style="list-style-type: none"> - Create test case for client - Perform test case - Record test result
Đỗ Đức Đạt	Tester	<ul style="list-style-type: none"> - Manage the testing result - Create test case for client - Conduct testing result

3.2 Test Environment

Phase	Tool	Provider	Version
Unit test	Manual	Manual	2.0.0
Integration test	Manual	Manual	2.0.0
System test	Manual	Manual	2.0.0
User acceptance test	Manual	Manual	2.0.1

3.3 Test Milestones

Milestone Task	Effort (Man days)	Start Date	End Date
User authentication	1	8/04/2025	9/04/2025
Account management			
Setting Management	0.5	9/04/2025	10/04/2025
Learner Management	1	10/04/2025	11/04/2025
Follower Management			
Question set management	1	11/04/2025	12/04/2025
Comment Management			
Learn Mode Interaction	2	12/04/2025	14/04/2025
Quiz Mode Interaction			
Question Set Progress Management			
Fields management	1	14/04/2025	15/04/2025
Subject Management			
Group management	1	15/04/2025	16/04/2025

Report Management	0.5	16/04/2025	17/04/2025
Goal Management	0.5	16/04/2025	17/04/2025
Additional Features	0.5	17/04/2025	18/04/2025
Statistic Interaction	0.5	17/04/2025	18/04/2025

4. Test Cases

- Unit Test Cases: [Report5_Unit Test.xlsx](#)
- Other Test Cases: [Report5_Test Report](#)

5. Test Reports

No	Function code	Passed	Failed	Untested	N	A	B	Total Test Cases
1	Function 1	9	0	0	4	5	0	9
2	Function 2	3	0	0	1	2	0	3
3	Function 3	10	0	0	3	6	1	10
4	Function 4	9	0	0	1	6	2	9
5	Function 5	5	0	0	1	4	0	5
6	Function 6	3	0	0	1	2	0	3
7	Function 7	2	0	0	1	1	0	2
8	Function 8	4	0	0	1	3	0	4
9	Function 9	3	0	0	1	2	0	3
10	Function 10	2	0	0	1	1	0	2
11	Function 11	2	0	0	1	1	0	2
12	Function 12	10	0	0	1	7	2	10
13	Function 13	10	0	0	1	7	2	10
14	Function 14	2	0	0	1	1	0	2
15	Function 15	2	0	0	1	1	0	2
16	Function 16	2	0	0	1	1	0	2
17	Function 17	2	0	0	1	1	0	2
18	Function 18	2	0	0	1	1	0	2
19	Function 19	2	0	0	1	1	0	2
20	Function 20	9	0	0	1	7	1	9
21	Function 21	2	0	0	1	1	0	2
22	Function 22	2	0	0	1	1	0	2
23	Function 23	2	0	0	1	1	0	2
24	Function 24	2	0	0	1	1	0	2
25	Function 25	2	0	0	1	1	0	2
26	Function 26	6	0	0	2	4	0	6
27	Function 27	6	0	0	2	4	0	6
28	Function 28	2	0	0	1	1	0	2
29	Function 29	2	0	0	1	1	0	2
30	Function 30	9	0	0	1	7	1	9

Figure 5.a: Unit test list - 1

31	Function 31	1	0	0	1	0	0	1
32	Function 32	1	0	0	1	0	0	1
33	Function 33	1	0	0	1	0	0	1
34	Function 34	2	0	0	1	1	0	2
35	Function 35	2	0	0	1	1	0	2
36	Function 36	2	0	0	1	1	0	2
37	Function 37	8	0	0	3	4	1	8
38	Function 38	9	0	0	3	5	1	9
39	Function 39	3	0	0	1	2	0	3
40	Function 40	7	0	0	3	4	0	7
41	Function 41	2	0	0	1	1	0	2
42	Function 42	2	0	0	1	1	0	2
43	Function 43	2	0	0	1	1	0	2
44	Function 44	10	0	0	1	7	2	10
45	Function 45	3	0	0	1	2	0	3
46	Function 46	1	0	0	1	0	0	1
47	Function 47	2	0	0	1	1	0	2
48	Function 48	2	0	0	1	1	0	2
49	Function 49	2	0	0	1	1	0	2
50	Function 50	1	0	0	1	0	0	1
51	Function 51	1	0	0	1	0	0	1
52	Function 52	2	0	0	1	1	0	2
53	Function 53	2	0	0	1	1	0	2
54	Function 54	2	0	0	1	1	0	2
55	Function 55	2	0	0	1	1	0	2
56	Function 56	1	0	0	1	0	0	1
57	Function 57	1	0	0	1	0	0	1
58	Function 58	1	0	0	1	0	0	1
59	Function 59	10	0	0	1	8	1	10
60	Function 60	10	0	0	1	8	1	10
61	Function 61	2	0	0	1	1	0	2
62	Function 62	1	0	0	1	0	0	1
63	Function 63	2	0	0	1	1	0	2
64	Function 64	2	0	0	1	1	0	2
65	Function 65	2	0	0	1	1	0	2
66	Function 66	7	0	0	3	4	0	7

Figure 5.b: Unit test list – 2

67	Function 67	1	0	0	1	0	0	1
68	Function 68	1	0	0	1	0	0	1
69	Function 69	2	0	0	1	1	0	2
70	Function 70	2	0	0	1	1	0	2
71	Function 71	9	0	0	1	7	1	9
72	Function 72	9	0	0	1	7	1	9
73	Function 73	2	0	0	1	1	0	2
74	Function 74	2	0	0	1	1	0	2
Sub total		267	0	0	89	161	17	267
	Test coverage	100,00 %						
	Test successful coverage	100,00 %						
	Normal case	33,33 %						
	Abnormal case	60,30 %						
	Boundary case	6,37 %						

Figure 5.c: Unit test list – 3



Figure 5.d: Unit test pie chart

No	Module code	Passed	Failed	Pending	N/A	Number of test cases
1	Authenticate Management	23	0	0	0	23
2	Account management	15	0	0	0	15
3	Question Set Management	43	0	0	0	43
4	Question Management	38	0	0	0	38
5	Comment Management	19	0	0	0	19
6	Learn Mode Interaction	7	0	0	0	7
7	Quiz Mode Interaction	9	0	0	0	9
8	Statistic Interaction	10	0	0	0	10
9	Additional features	16	0	0	0	16
10	Group management	9	0	0	0	9
11	Goal management	19	0	0	0	19
12	Setting Management	15	0	0	0	15
13	Learner management	20	0	0	0	20
14	Report management	12	0	0	0	12
15	Field management	24	0	0	0	24
16	Subject Management	24	0	0	0	24
17	Question set progress management	2	0	0	0	2
18	Follower Management	9	0	0	0	9
Sub total		305	0	0	0	305
Test coverage			100.00	%		
Test successful coverage			100.00	%		

Figure 5.e: System test list

Phase	Completion criteria	Criteria met
Unit Test	All tests execute successfully with full coverage achieved.	Met
Integration Test	Passes without any defects or issues that need to be resolved.	Met
System Test	The system satisfies all the defined functional and non-functional requirements.	Met
User Acceptance Test	The system meets the initial requirements.	Met

VI. Release Package & User Guides

1. Deliverable Package

No.	Items	Sub-Items	Type	Version
<i>Code Package</i>				
1	Online learning platform - Client	Online learning platform - website	New	2.0
2	Online learning platform - Server	Online learner platform - API endpoints	New	2.0
<i>Database</i>				
1	Tables & Datas	seed.ts	New	3.0
<i>Documents</i>				
1	Report 1	Project Introduction	New	1.0
2	Report 2	Project Management Plan	New	1.0
3	Report 3	Software Requirement Specification	New	1.0
4	Report 4	Software Design description	New	1.0
5	Report 5	Software Testing documentation	New	1.0
6	Report 6	Release Package & User Guides	New	1.0
7	Report 7	Final Report	New	1.0
<i>Testing</i>				
1	Unit test	Report5_Unit Test.xlsx	New	1.0
2	System test	Report5_Test Report	New	1.0

2. Installation Guides

2.1 System Requirements

- For server:
 - Operating system: Windows 10 or higher
 - Storage: SSD 256GB or higher.
 - RAM: 8GB or higher.
 - CPU: Intel Core i5 8th generation or AMD Ryzen 5 3600 or higher.
 - Database: MongoDB Compass 1.45.4
 - Runtime environment: Node.js 18.17 or higher
- For web client:
 - Operating system: Windows 10 or higher.
 - Browser: Google Chrome, Safari (macOS, IOS), Edge, Brave, Opera, Firefox, etc.
 - Runtime Environment: Node.js 18.17 or higher

2.2 Setup Files

- Quizzzee-client.exe
- Quizzzee-server.exe

2.3 Installation Instruction

2.3.1 Install Node.js

- Go to <https://nodejs.org/en>

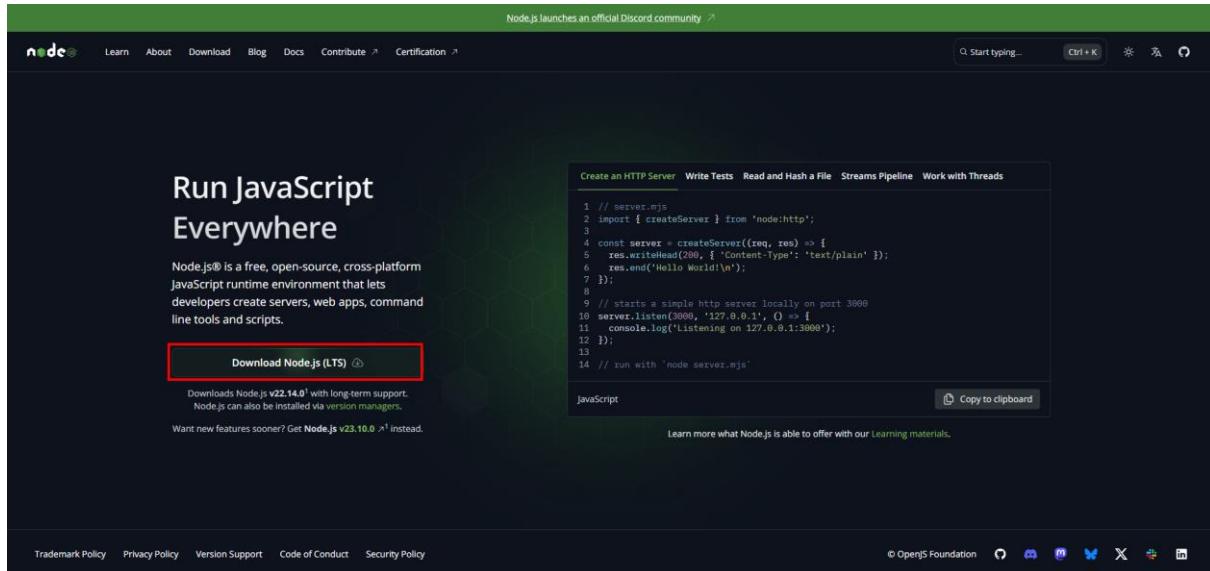


Figure 2.3.1.1: Install Node.js

- Click “Download Node.js (LTS)” and install the installer

- Run the installer to setup

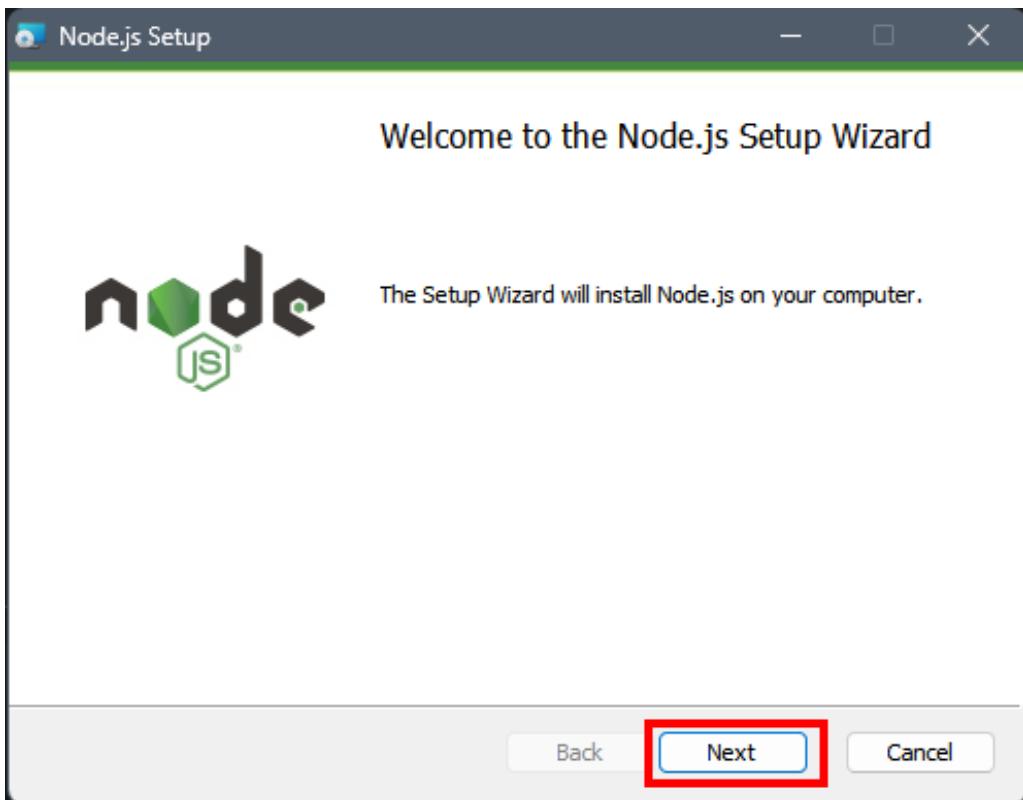


Figure 2.3.1.2: Install Node.js - 2

- Click "Next"



Figure 2.3.1.3: Install Node.js - 3

- (1) Check the “I accept the terms in the License Agreement” check box
- (2) Click “Next”

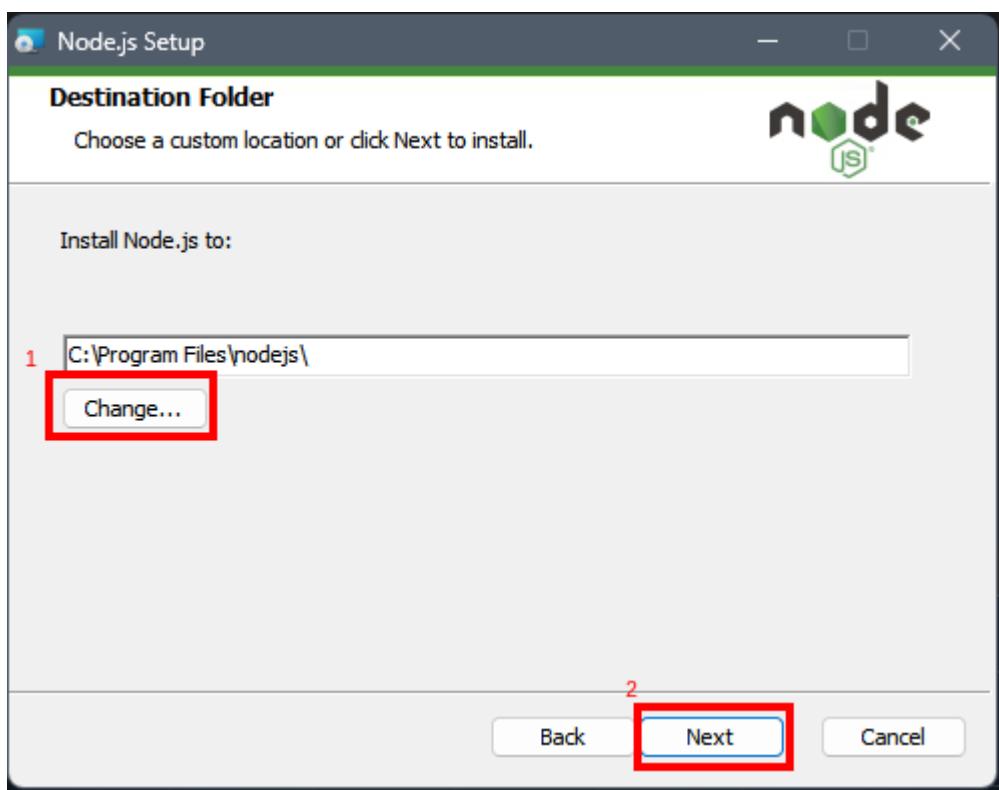


Figure 2.3.1.4: Install Node.js - 4

- (1) Choose a destination folder
- (2) Click “Next”

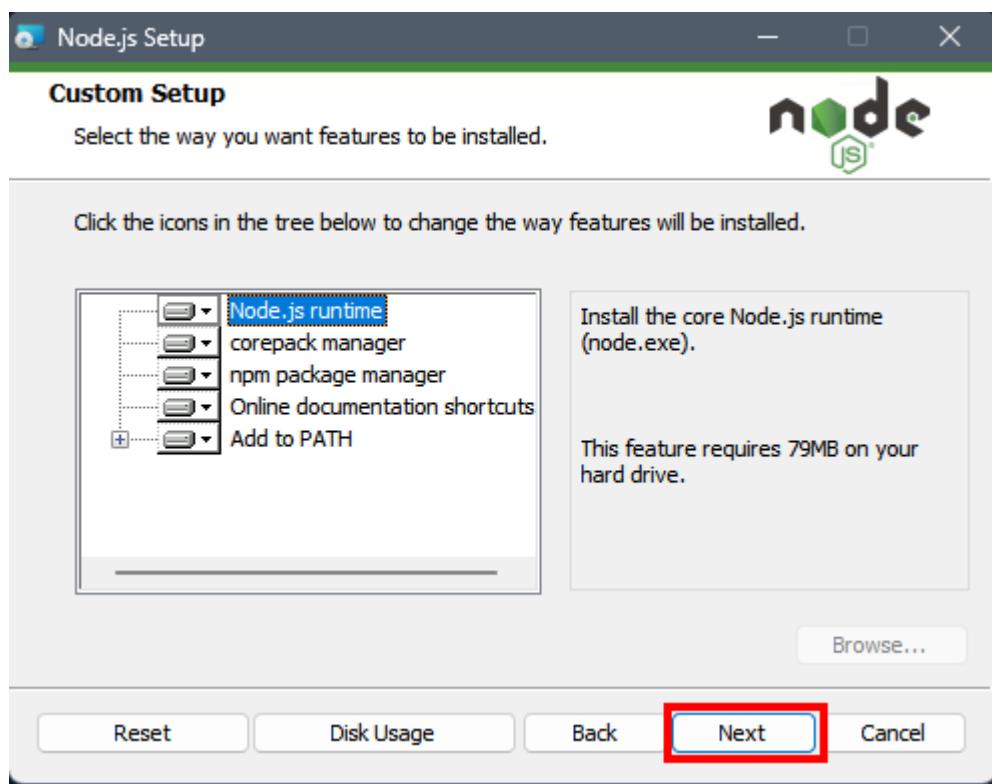


Figure 2.3.1.5: Install Node.js - 5

- Click “Next”

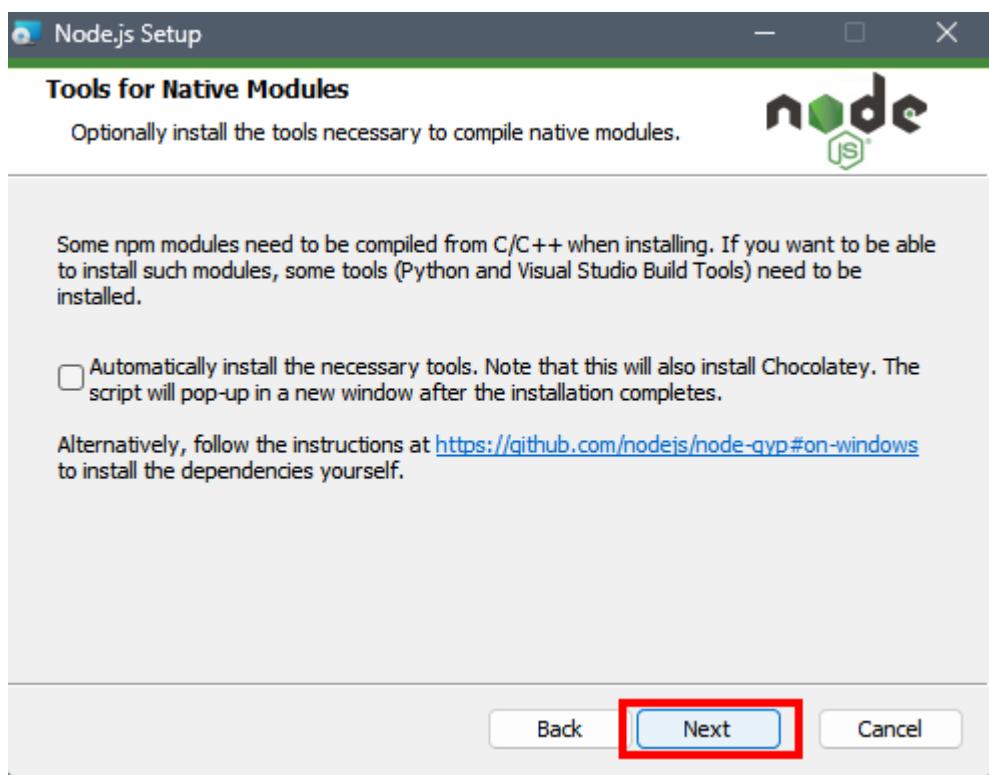


Figure 2.3.1.6: Install Node.js - 6

- Click “Next”

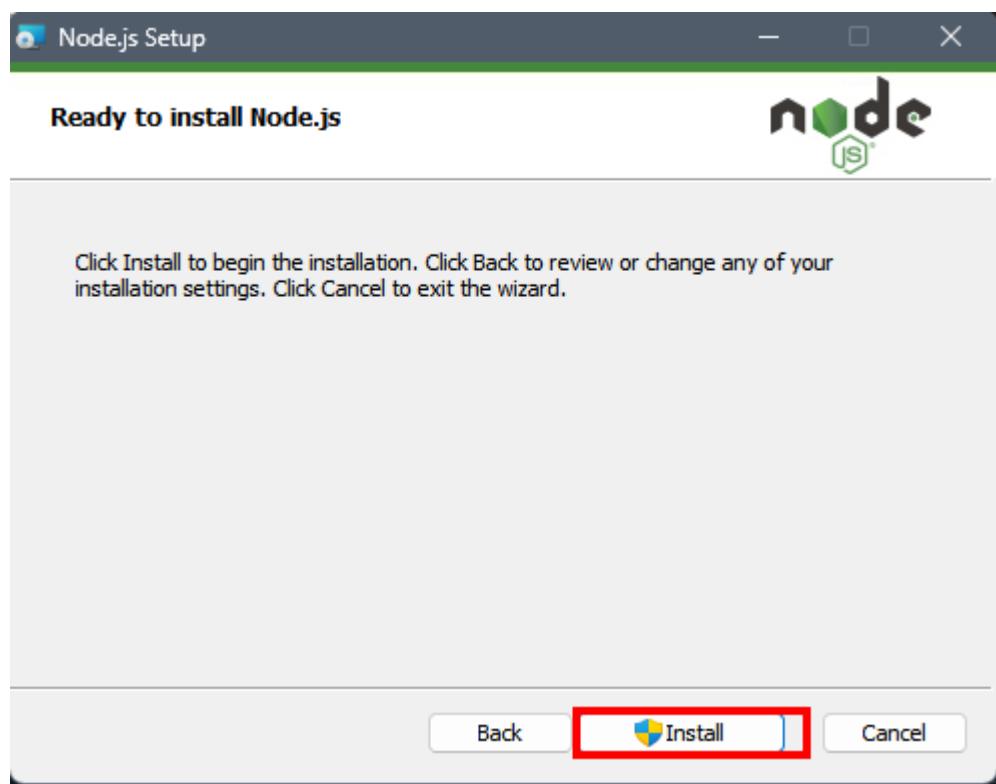


Figure 2.3.1.7: Install Node.js - 7

- Click “Install”

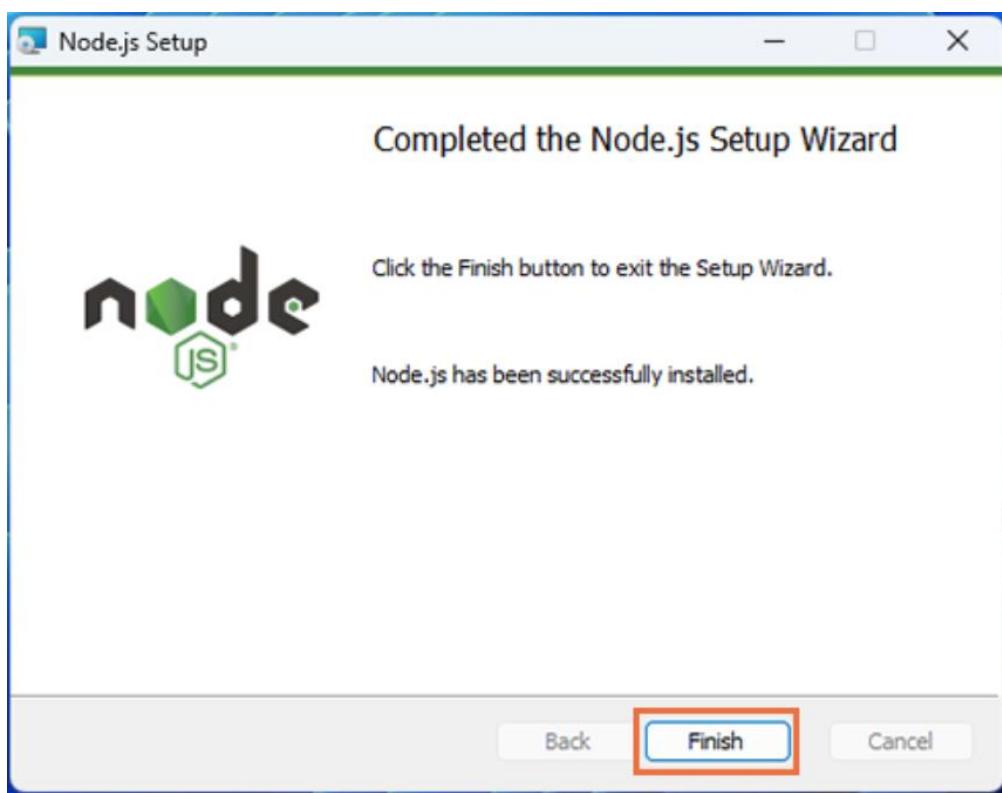


Figure 2.3.1.8: Install Node.js - 8

- Once the installation is complete, click "Finish"

2.3.2 Install MongoDB and MongoDB Compass

- Go to <https://www.mongodb.com/try/download/community>

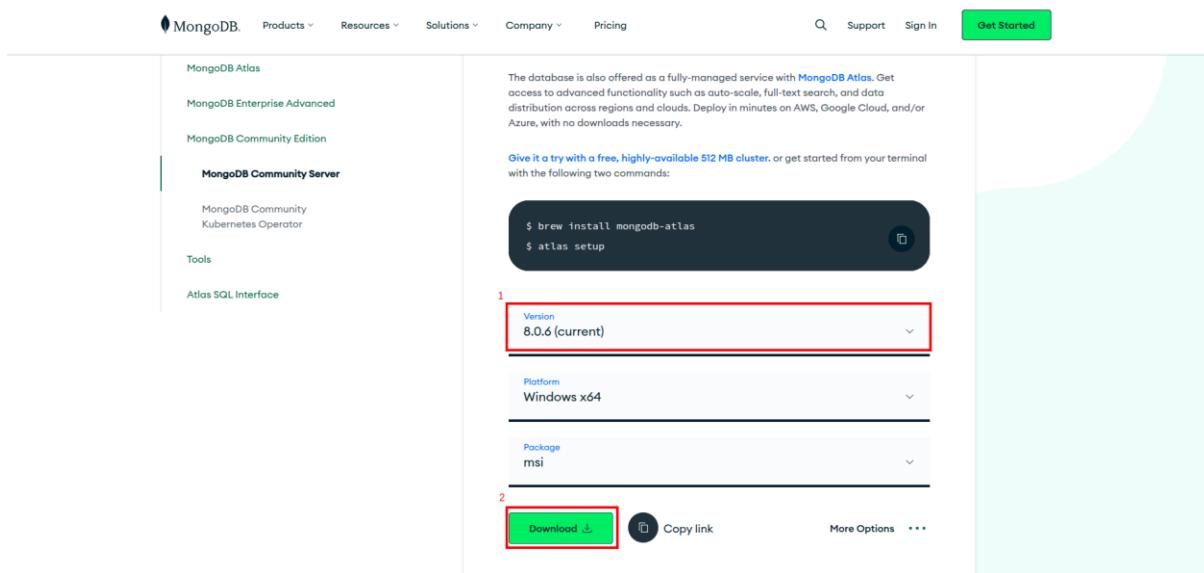


Figure 2.3.2.1: Install MongoDB and MongoDB Compass

- (1) Choose a mongoDB version
- (2) Click “Download”

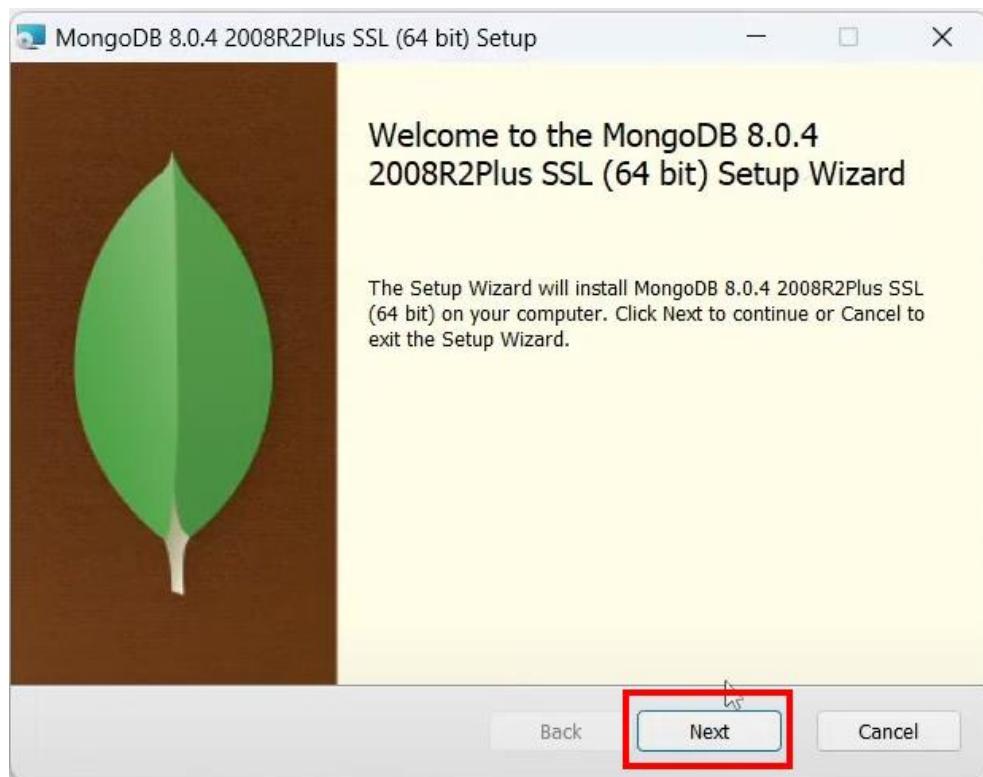


Figure 2.3.2.2: Install MongoDB and MongoDB Compass - 2

- Click “Next”

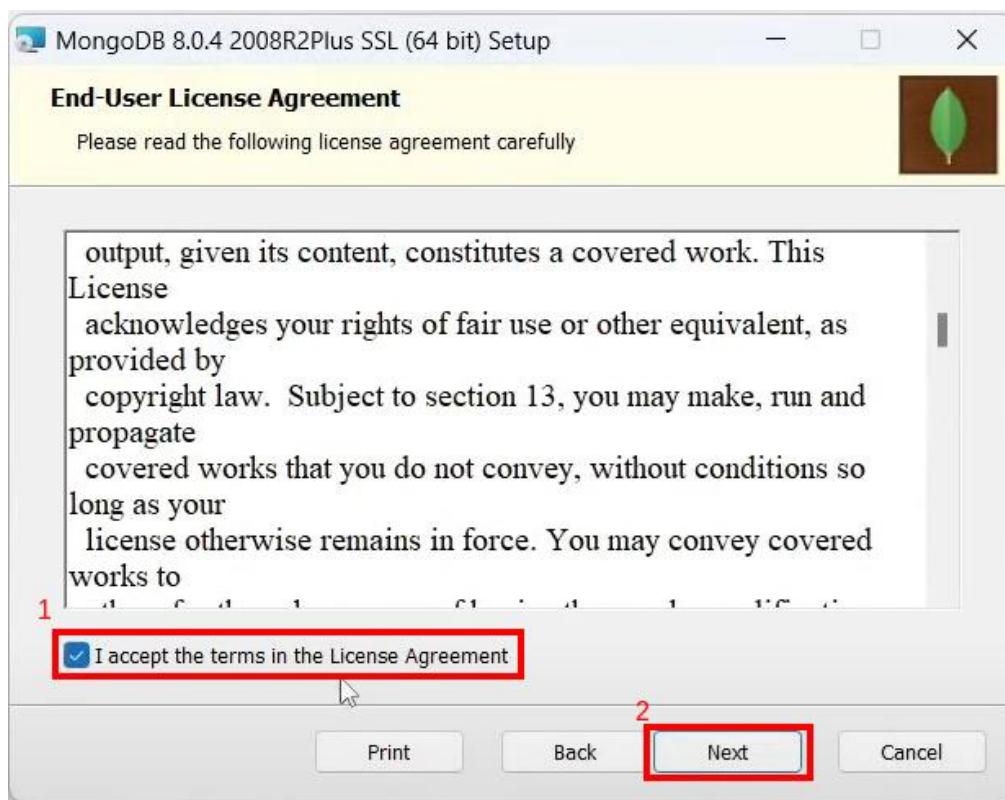


Figure 2.3.2.3: Install MongoDB and MongoDB Compass - 3

- (1) Check the “I accept the terms in the License Agreement” checkbox
- (2) Click “Next”

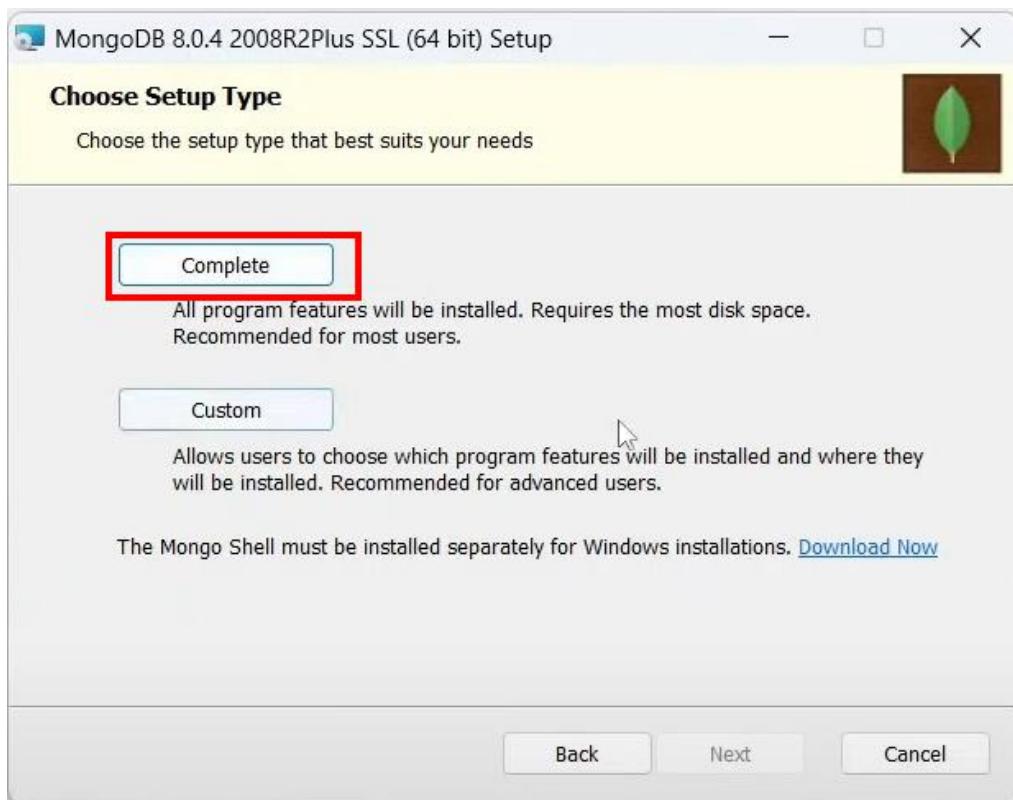


Figure 2.3.2.4: Install MongoDB and MongoDB Compass - 4

- Click “Complete”

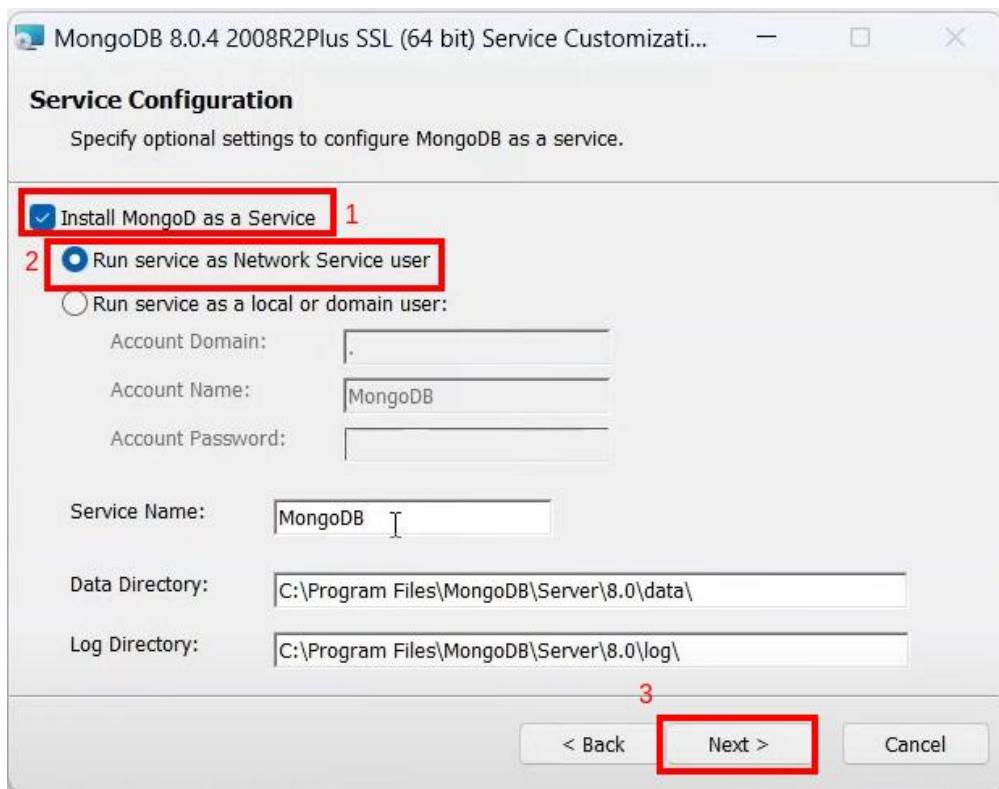


Figure 2.3.2.5: Install MongoDB and MongoDB Compass - 5

- (1) Check the “Install MongoDB as a Service” checkbox
- (2) Choose “Run service as Network Service user”
- (3) Click “Next”

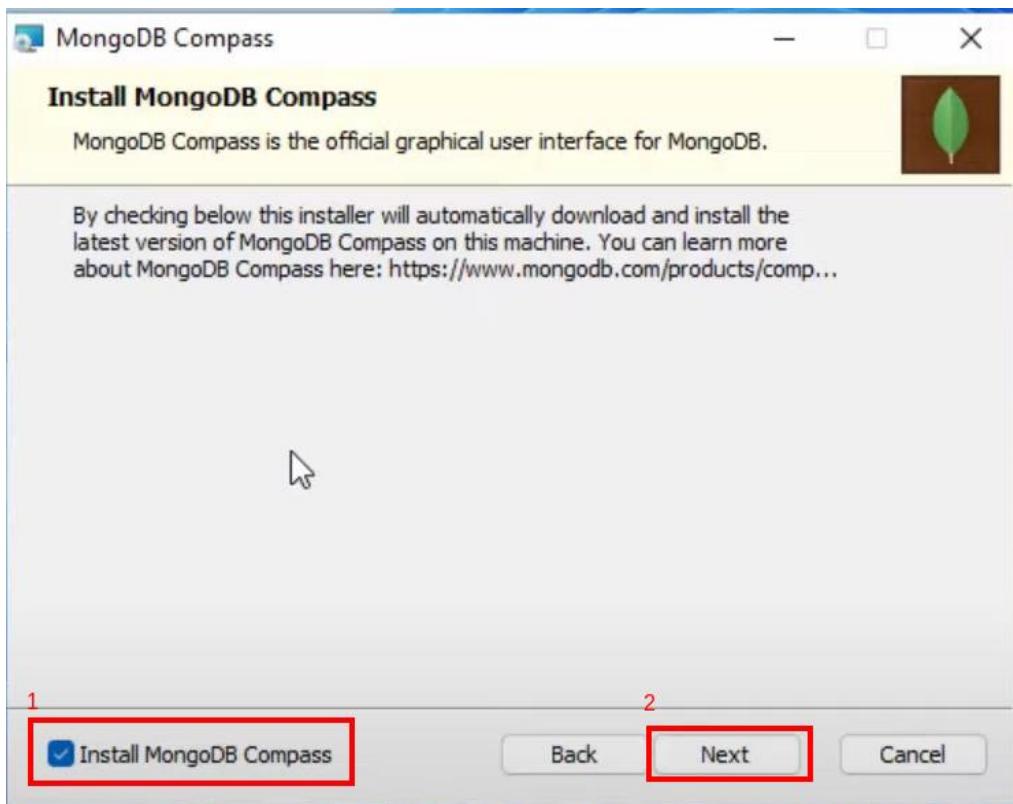


Figure 2.3.2.6: Install MongoDB and MongoDB Compass - 6

- (1) Check the “Install MongoDB Compass” checkbox
- (2) Click “Next”

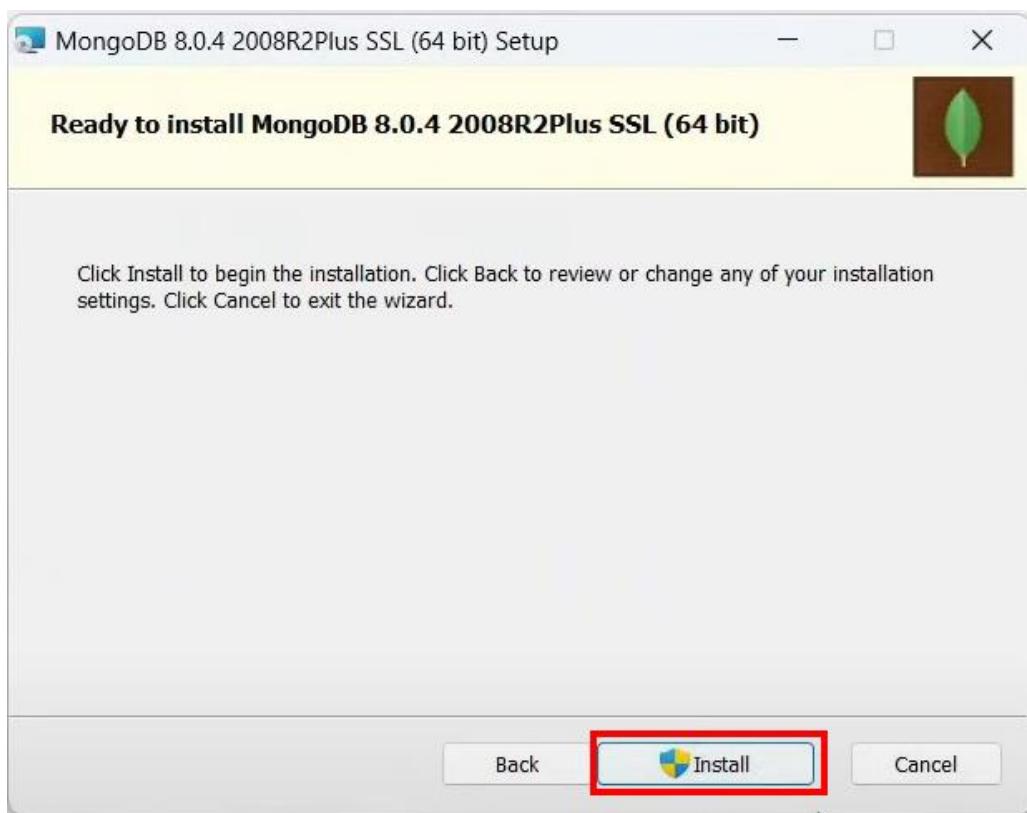


Figure 2.3.2.7: Install MongoDB and MongoDB Compass - 7

- Click “Install”

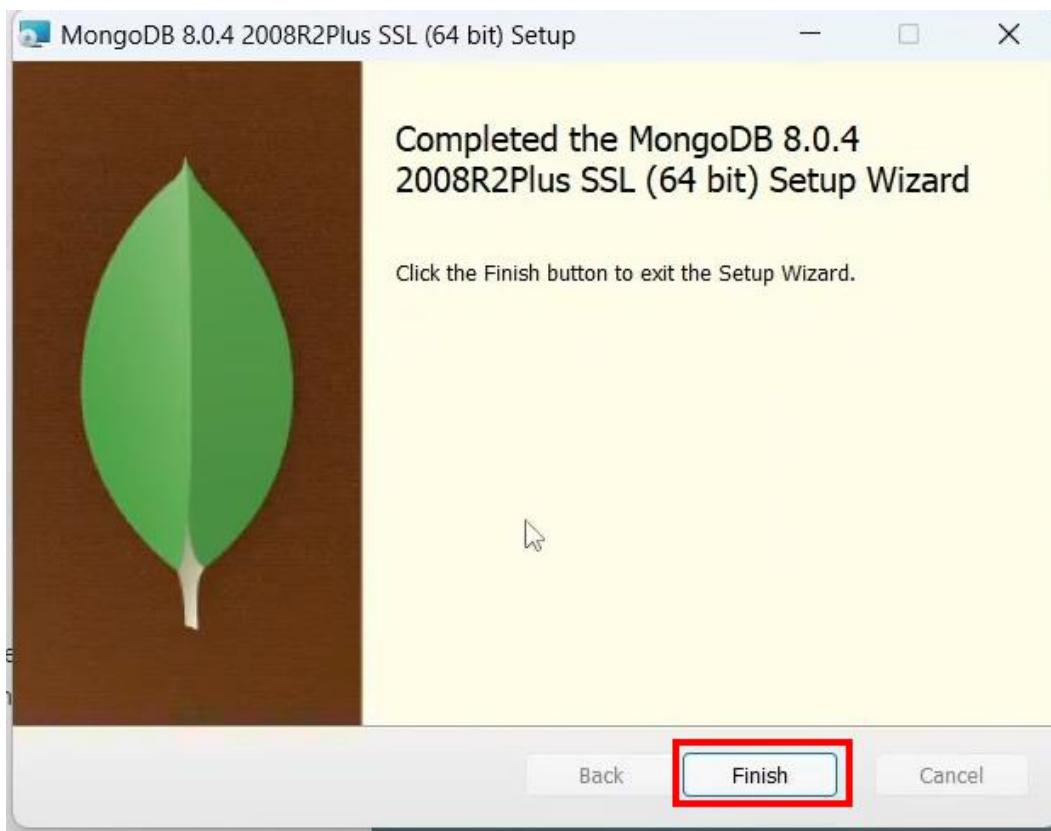


Figure 2.3.2.8: Install MongoDB and MongoDB Compass - 8

- Once the installation is complete, click “Finish”

In order to run the system, you will need to setup a replica set for MongoDB, follow to the official link below for setup: <https://www.mongodb.com/docs/manual/tutorial/convert-standalone-to-replica-set/>

3. User Manual

3.1 Terms and definitions

No.	Term	Definition
1	Responsive Design	A design approach that ensures the Online Learning Platform adapts and displays optimally on various screen sizes.
2	User Authentication	The process of verifying user identities, typically through usernames and passwords, to grant access to their accounts and personalised content.
3	Memory and Performance Optimization	Strategies and techniques implemented to efficiently use system resources, reduce data transfer times through caching mechanisms, and ensure smooth playback even during periods of high traffic.
4	Analytics Tools	Software and systems used to track and analyse user behaviour and preferences, aiding in the delivery of personalised content recommendations.

3.2 System requirements

Host Operating Systems:

The website should be compatible with the following host operating systems:

- Windows
- macOS
- Linux

Configurations:

The website's configurations should meet the following requirements:

- Responsive design for optimal viewing on any screen size.
- Implementation of secure authentication and user account management.

Memory and Performance:

The website's performance should be optimised for the following memory and performance considerations:

- Efficient use of system resources to minimise server load.
- Caching mechanisms to reduce data transfer and enhance load times.
- Regular monitoring and optimization for smooth response even during peak traffic.

3.3 Application Usage

3.3.1. Overview

Our system has 3 roles (Administrator, learner, guest). All the users can perform the following features:

- View question sets
- View question set detail
- View comments
- View groups
- View group detail
- Update system appearance
- View fields
- View subjects
- View learners
- View learner detail

The Administrator can perform these following features:

- Sign in
- Sign out
- Profile management (view, update, reset password)
- Learner management (block, unblock)
- Question set management (block, unblock)
- Group management (block, unblock)
- Statistics interaction (view, filter)
- Field management (view, create, update, delete)
- Subject management (view, create, update, delete)
- Report management (view, update)

Guests can perform these following features:

- Sign up
- Question set interaction (share, report, filter)
- Question set comment interaction (report, filter)
- Use learn mode
- Use matching mode
- Use quiz mode
- Use Pomodoro timer
- View quotes

Learners can perform these features:

- Sign in
- Sign out
- Profile management (view, update, reset password, delete)
- Following management (follow, unfollow)
- Question set management (create, update, delete)
- Question set interaction (share, export, report, filter, favourite)
- Question set contribution (rate, interact with question comment)
- Question set comment management (filter, mark, create, reply, update, delete, upvote, report)
- Use learn mode
- Use matching mode
- Use quiz mode
- Question set tracking (revisit quiz, view quiz statistics, view progress)
- Group management (create, update, delete, send invitation, join, leave)
- Goal management (view, create, update, delete)
- Settings management (view, update, reset)
- Use Pomodoro timer
- View achievements
- View quotes

3.3.2. Sign in

3.3.2.1. Administrator and Learner sign in with credentials

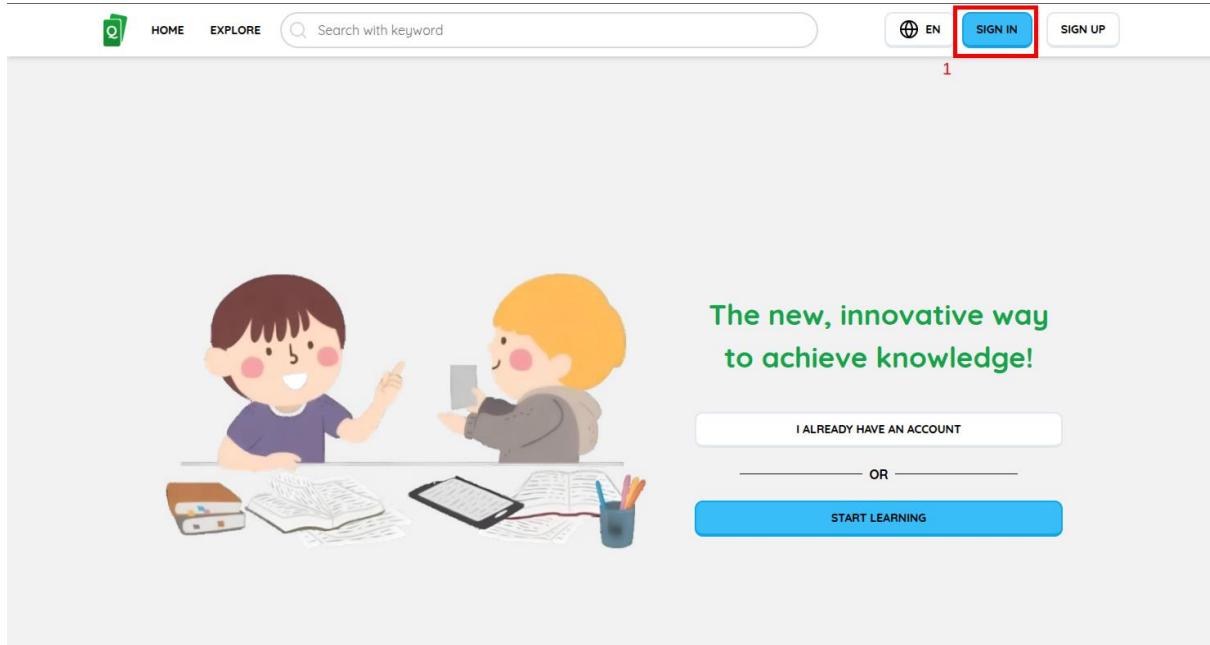


Figure 3.3.2.1.a: Sign in with credentials - 1

- (1) User clicks on the sign in button

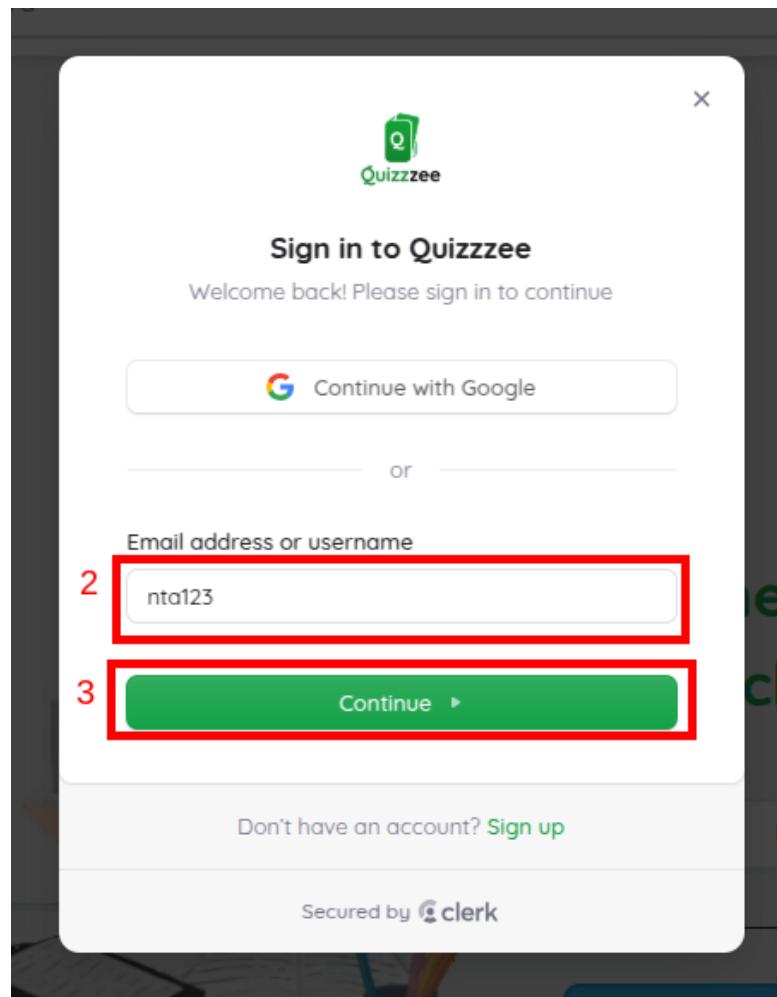


Figure 3.3.2.1.b: Sign in with credentials - 2

- (2) User enters their username or email
- (3) User clicks Continue button

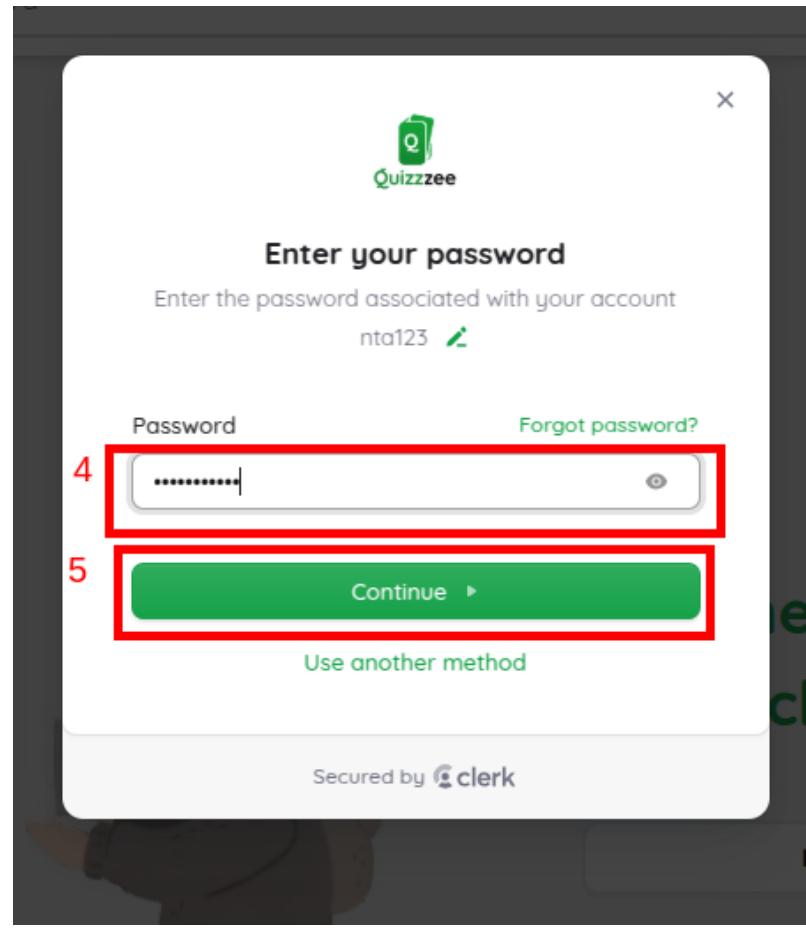


Figure 3.3.2.1.c: Sign in with credentials - 3

- (4) User enters password
- (5) User Clicks Continue button to complete sign in

3.3.2.2. Administrator and Learner sign in with OTP

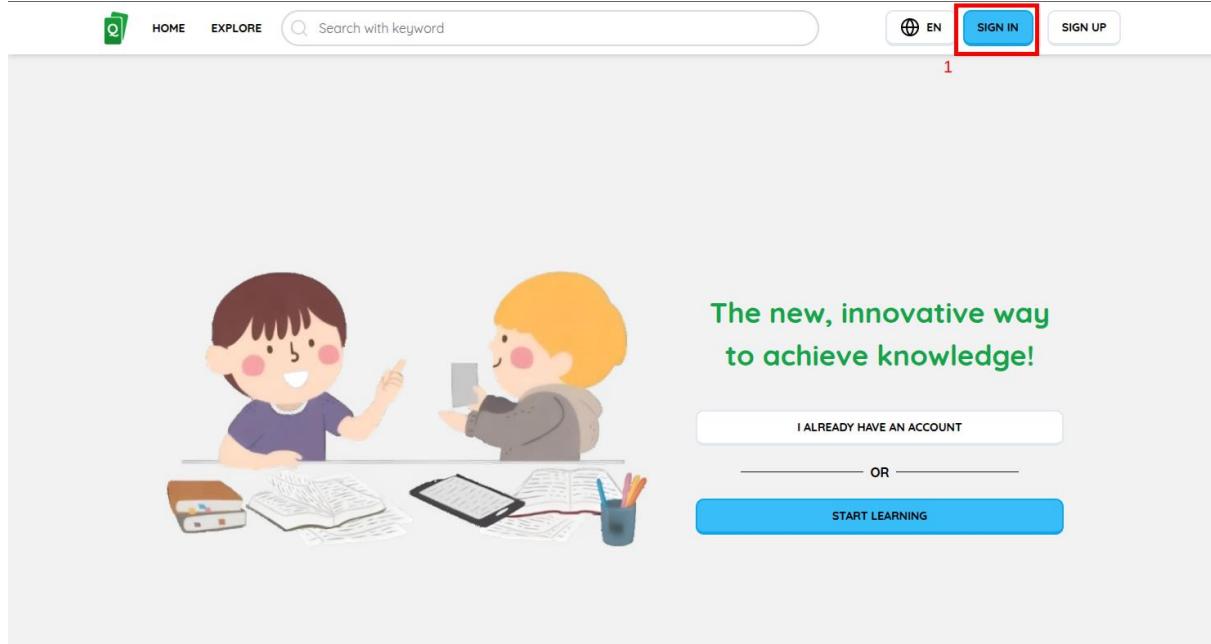


Figure 3.3.2.2.a: Sign in with OTP - 1

- (1) User clicks on the sign in button

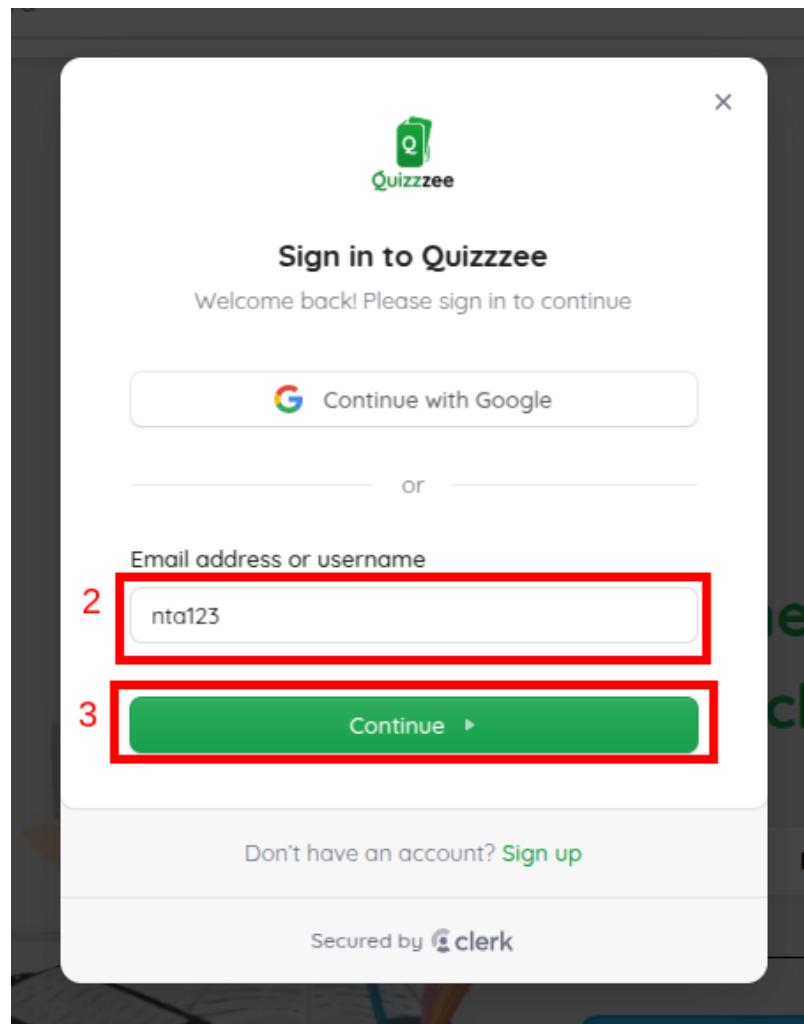


Figure 3.3.2.2.b: Sign in with OTP - 2

- (2) User enters their username or email
- (3) User clicks Continue button

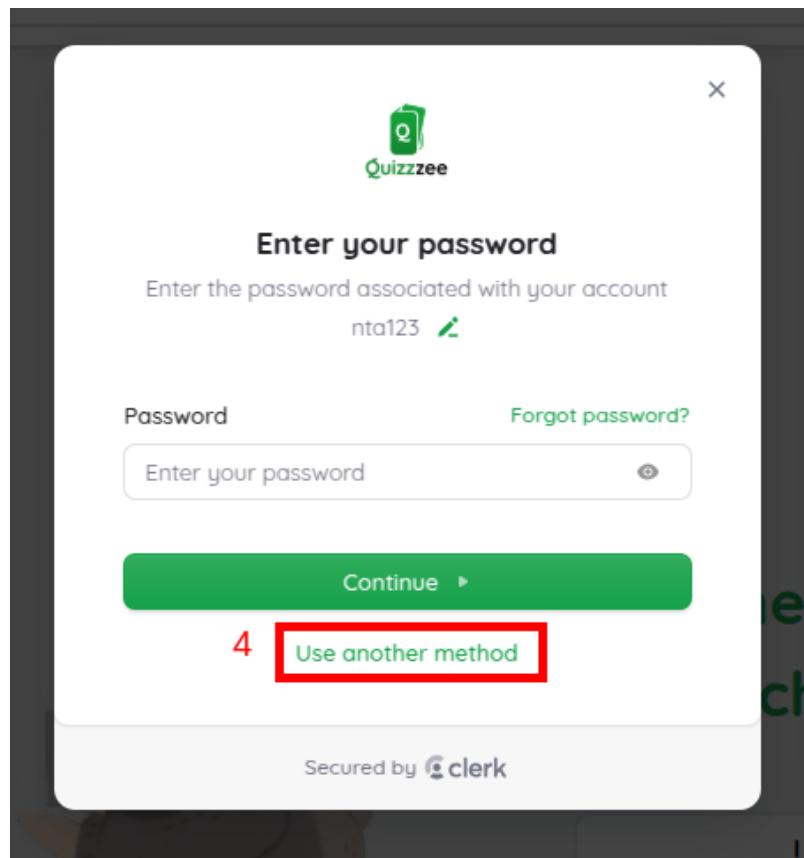


Figure 3.3.2.2.c: Sign in with OTP - 3

- (4) User clicks Use another method

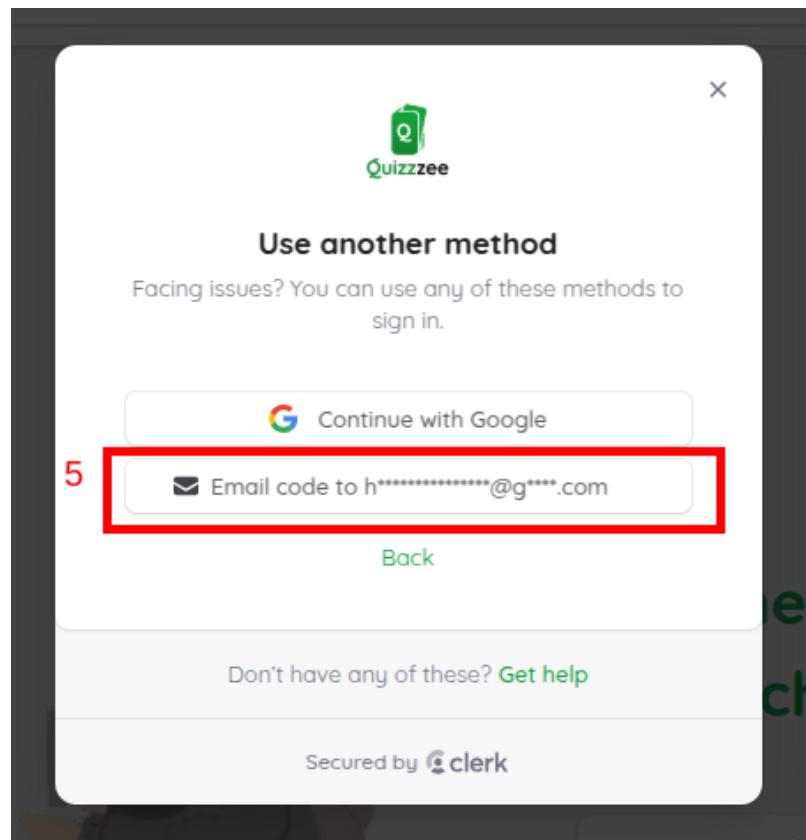


Figure 3.3.2.2.d: Sign in with OTP - 4

- (5) User clicks the send code button to receive an OTP sent to their email

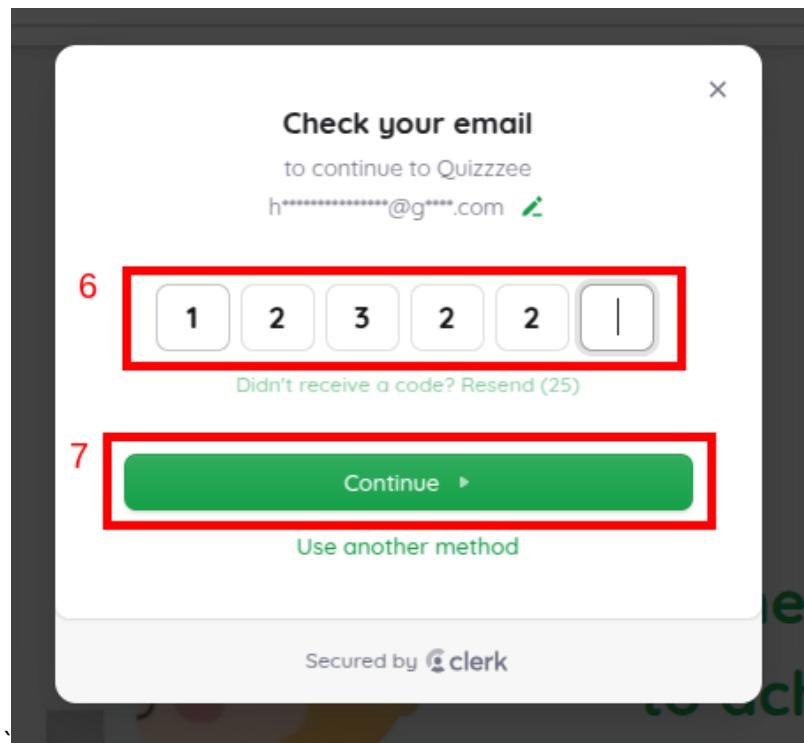


Figure 3.3.2.2.e: Sign in with OTP - 5

- (6) User enters the code.
- (7) User clicks Continue button to complete sign in

3.3.2.3. Administrator and Learner sign in with Google

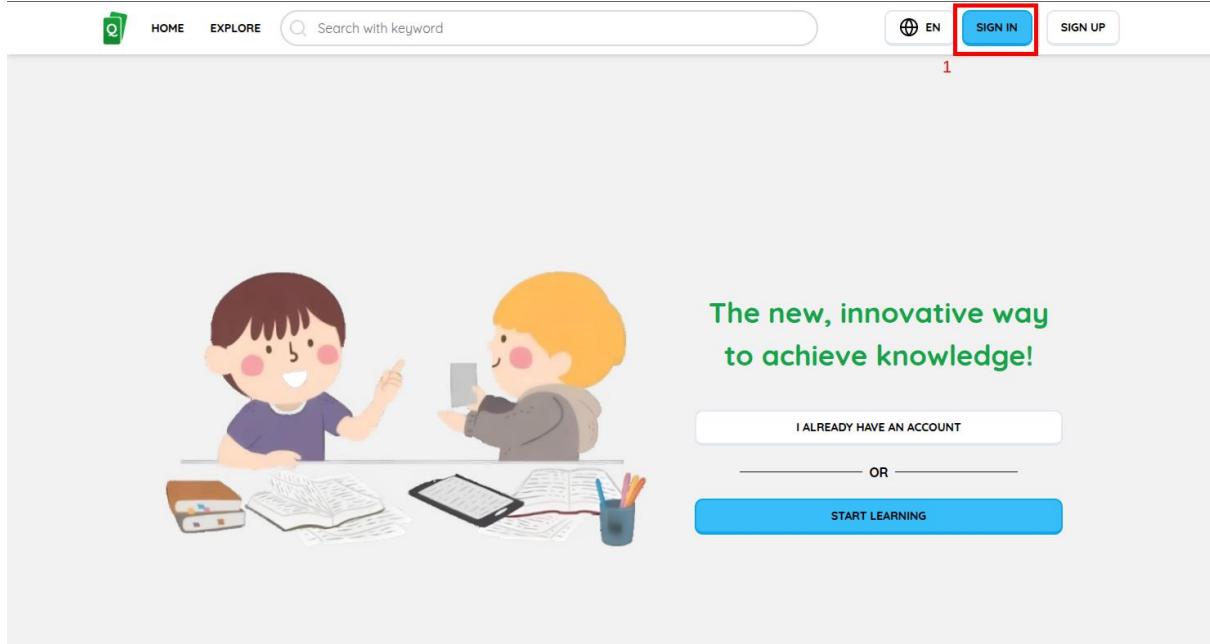


Figure 3.3.2.3.a: Sign in with Google - 1

- (1) User clicks on the sign in button

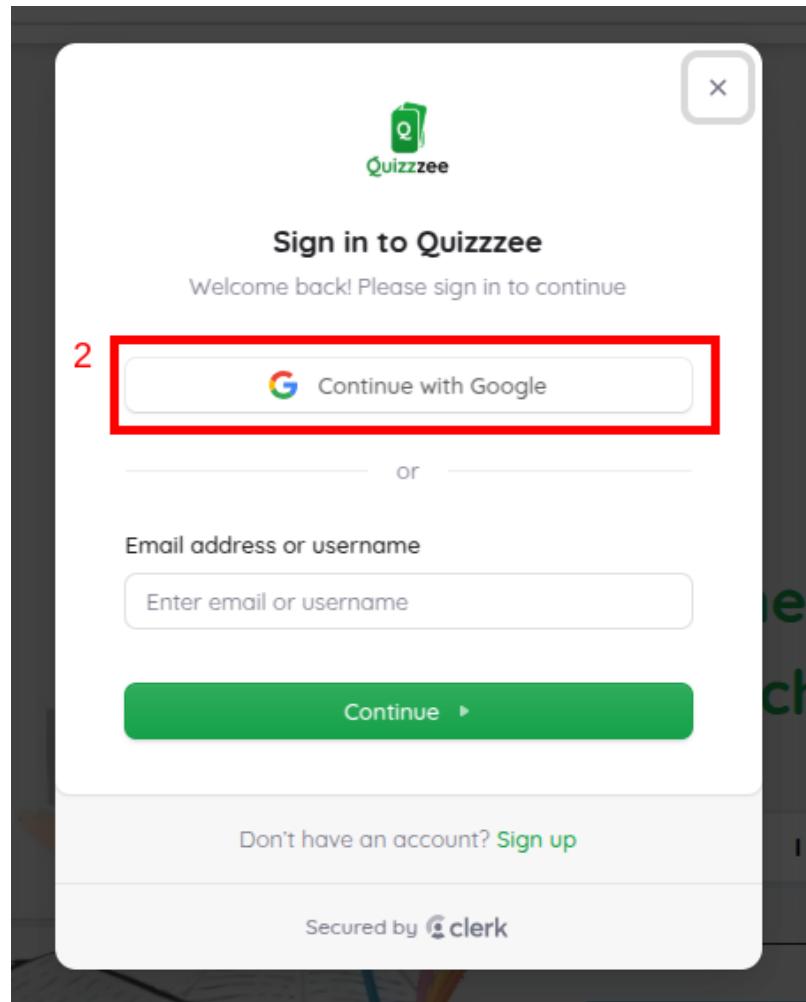


Figure 3.3.2.3.b: Sign in with Google – 2

- (2) User clicks on Continue with Google button

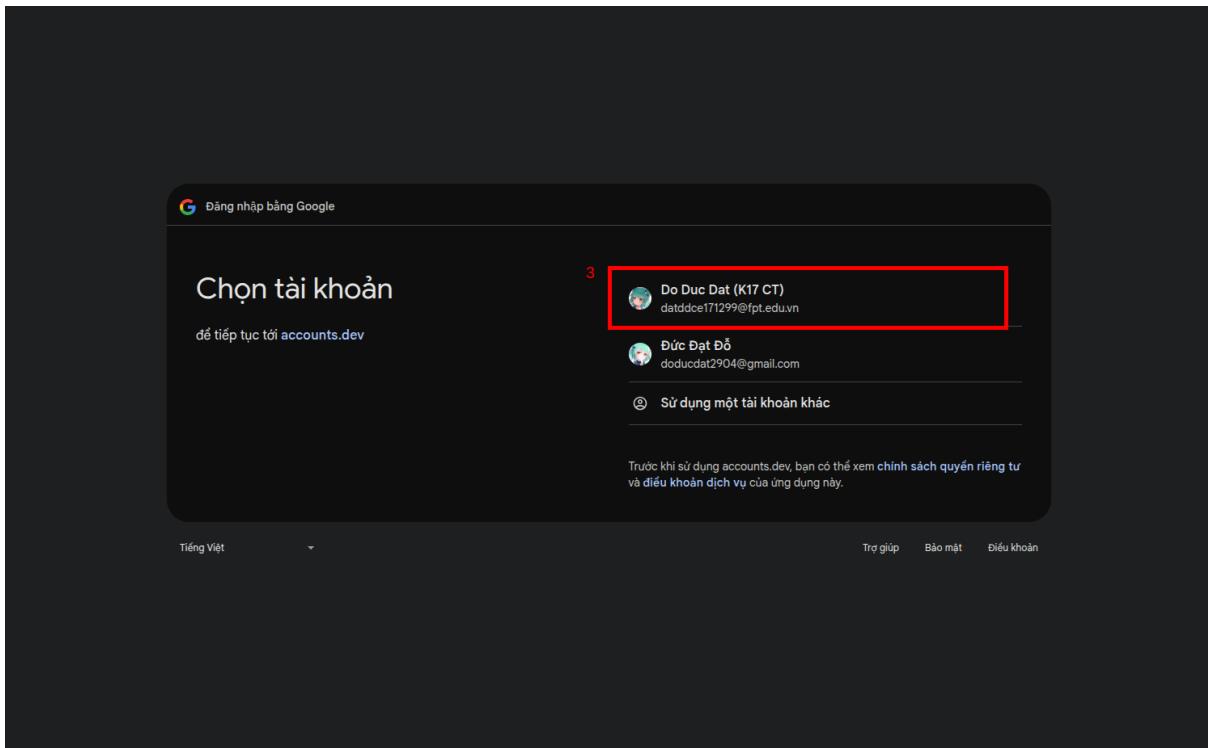


Figure 3.3.2.3.c: Sign in with Google – 3

- (3) User chooses a Google account to complete sign in

3.3.3. Sign up

3.3.3.1. Guest signs up with credential

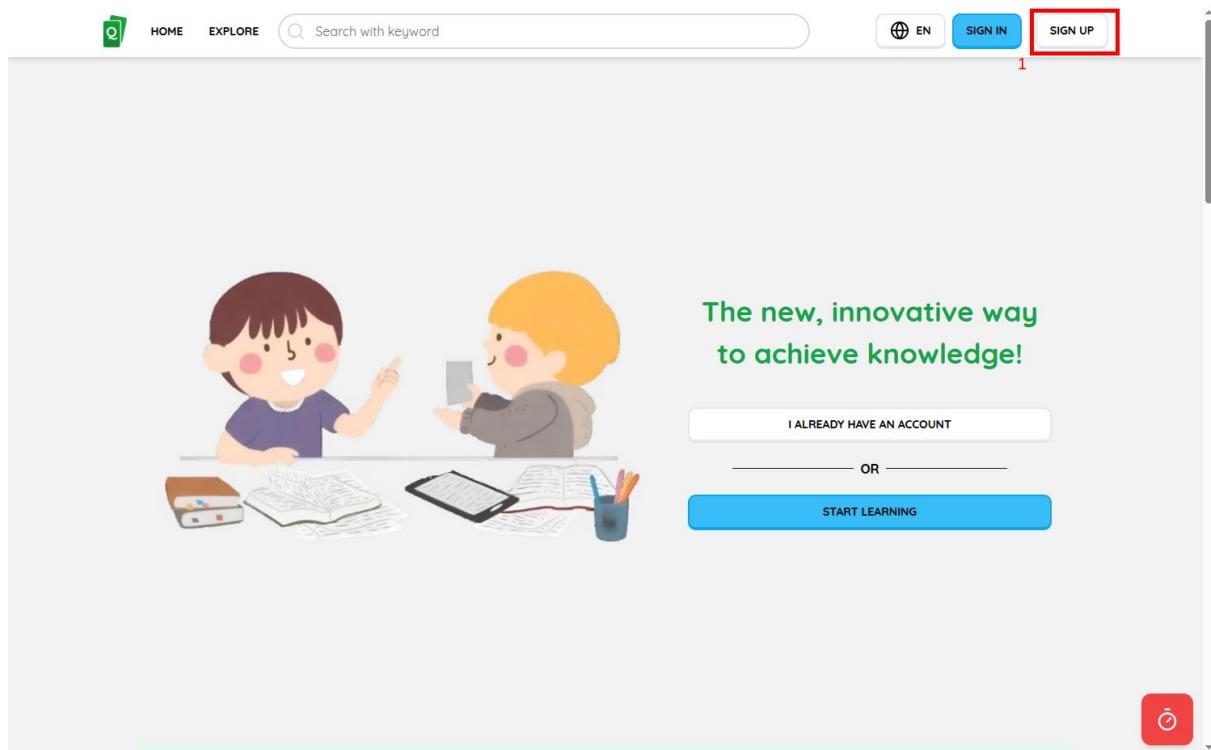


Figure 3.3.3.1.a: Sign up with credentials - 1

- (1) User clicks on the sign up button

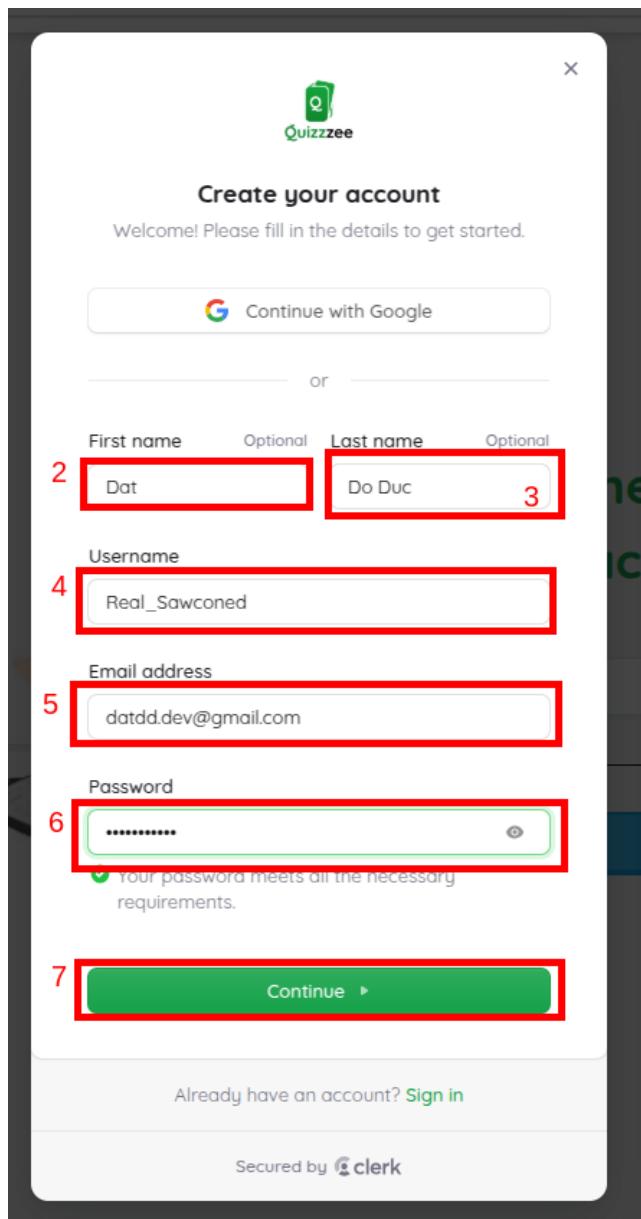


Figure 3.3.3.1.b: Sign up with credentials - 2

- (2) User enters first name (optional)
- (3) User enters last name (optional)
- (4) User enters username
- (5) User enters email address
- (6) User enters password
- (7) User clicks Continue button

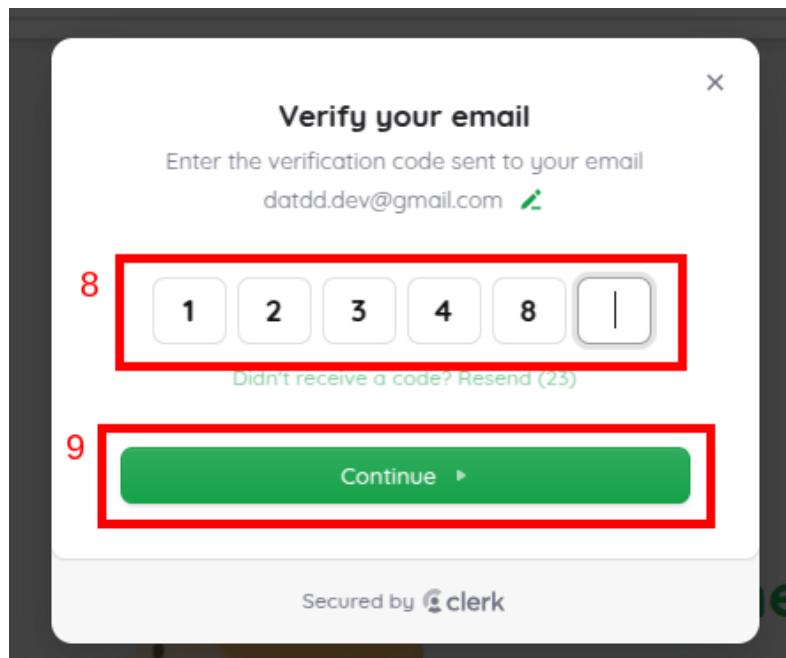


Figure 3.3.3.1.c: Sign up with credentials - 3

- (8) User enters code
- (9) User clicks continue button to complete sign up

3.3.3.2. Guest sign up with Google

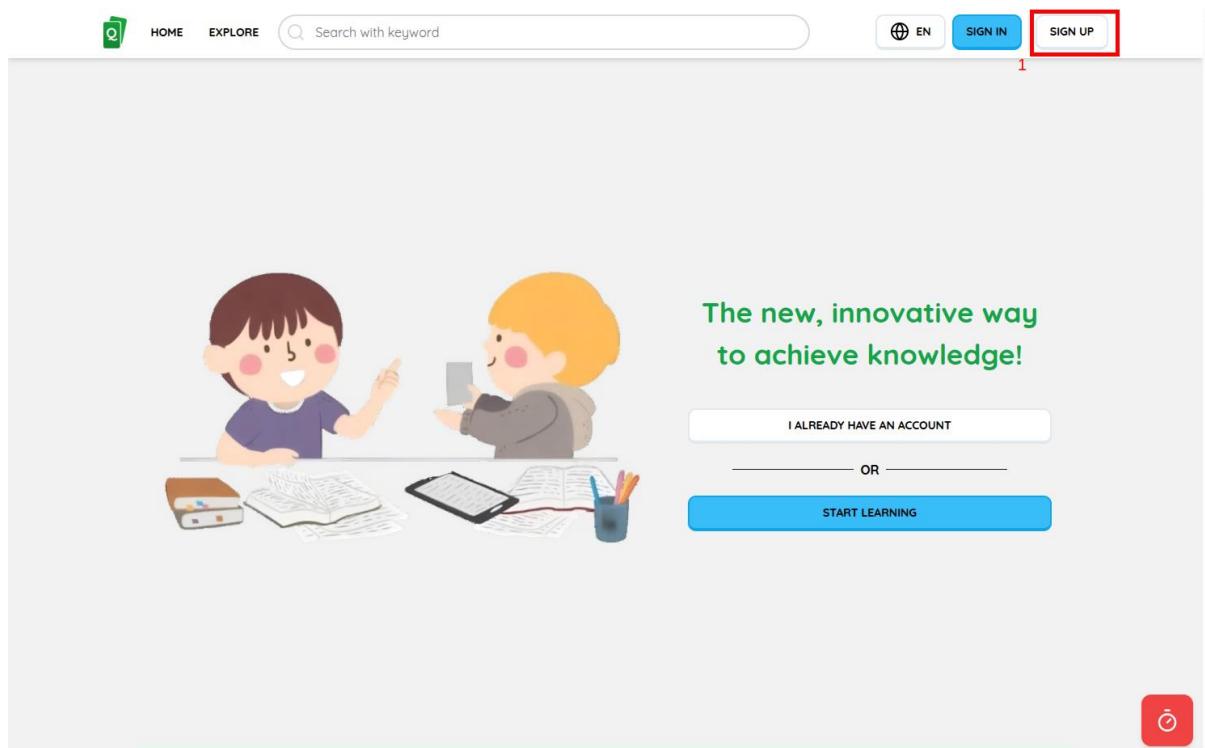


Figure 3.3.3.2.a: Sign up with Google - 1

- (1) User clicks on the sign up button

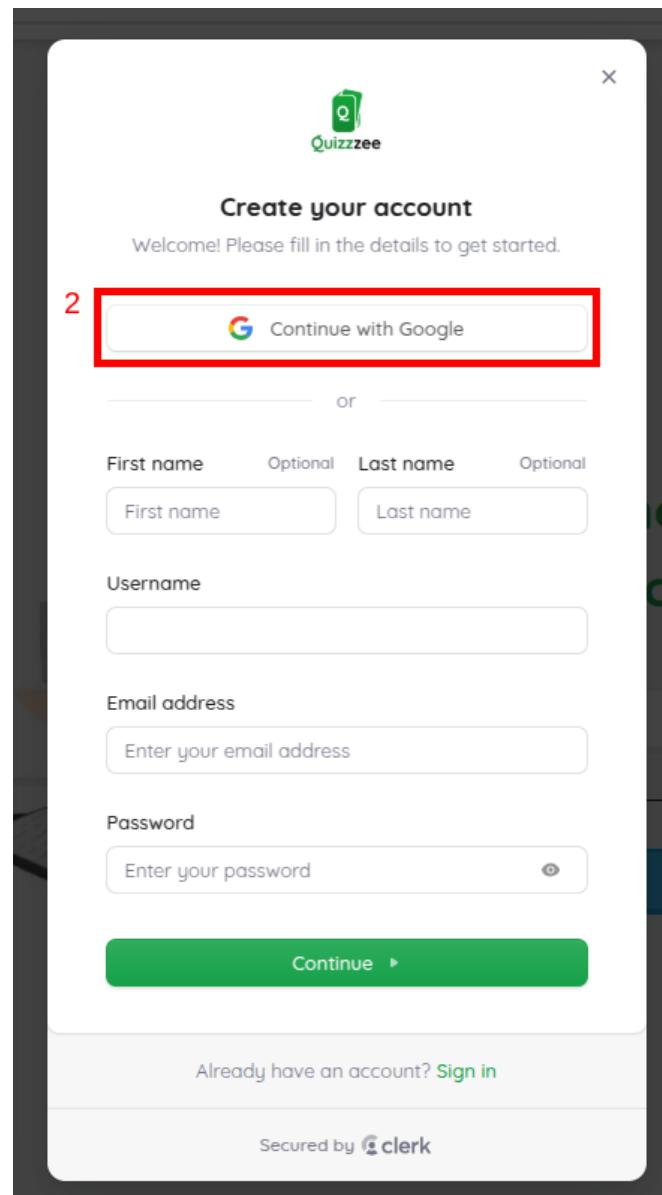


Figure 3.3.3.2.b: Sign up with Google - 2

- (2) User clicks Continue with Google button

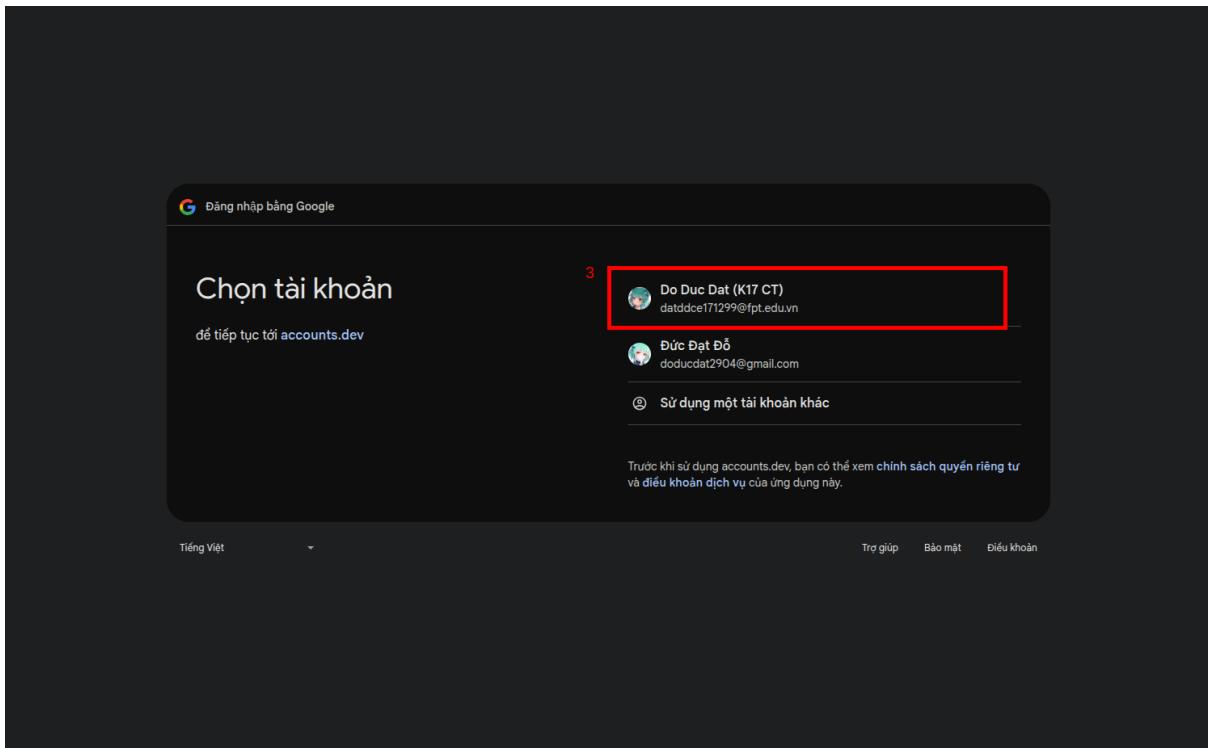


Figure 3.3.3.2.c: Sign up with Google - 3

- (3) User chooses a Google account to complete sign up

3.3.4 Sign out

3.3.4.1 Learner sign out from header

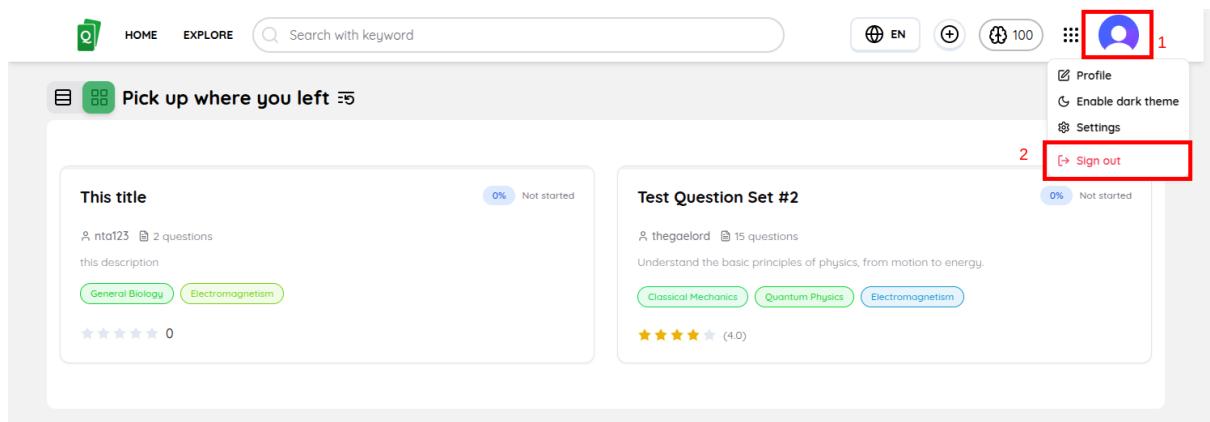


Figure 3.3.4.1 Sign out

- (1) User clicks on the profile image.
- (2) User clicks on the sign out button to complete signing out.

3.3.4.2 Learner sign out from the settings page

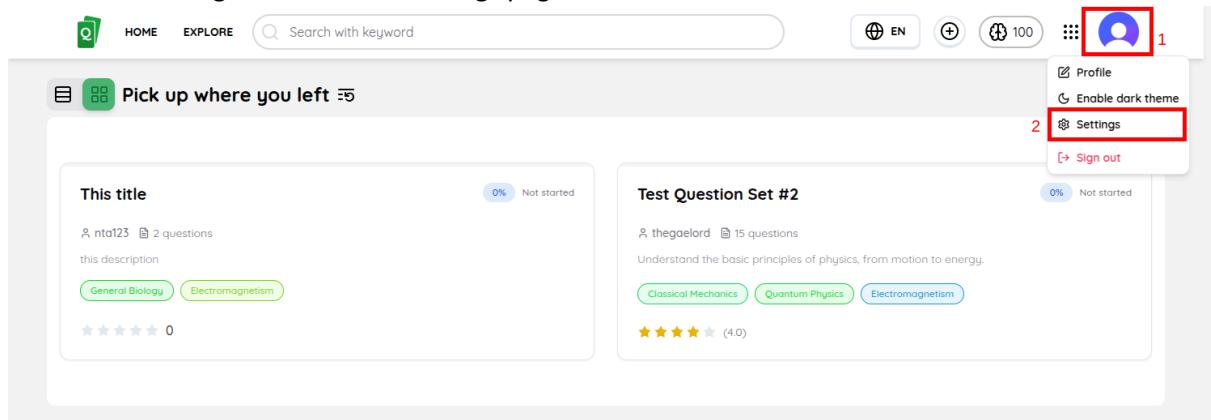


Figure 3.3.4.2.a: Sign out - 1

- (1) User clicks on the profile image
- (2) User clicks on the settings button to be redirected to the settings page.

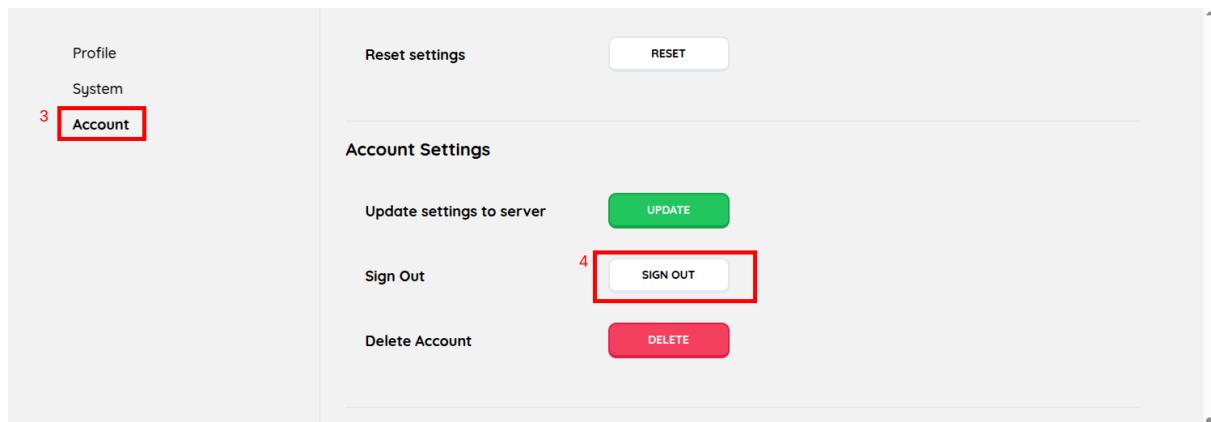


Figure 3.3.4.2.b: Sign out - 2

- (3) User scrolls down to the sign out section.
- (4) User clicks sign out to complete signing out.

3.3.4.3 Admin sign out from the sidebar

The screenshot shows the Quizzzee application interface. On the left, a sidebar displays navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. Below these are language and theme settings: EN (selected), Dark theme, and a user icon with the name nt0123. A red box highlights the 'Sign out' button next to the user icon, labeled '1'. Another red box highlights the user icon itself, labeled '2'. The main content area is titled 'Question Sets' and shows a table of 10 sets found. The table columns are TITLE, Description, Public, Banned, Author, Subject, and Date. The data is as follows:

TITLE	Description	Public	Banned	Author	Subject	Date
Test Question Set #1	Explore fundamental concepts in mathematics, including algebra and geometry.	Yes	No	thegaelord	Calculus,Linear Algebra,Statistics	May 4, 2025
Test Question Set #2	Understand the basic principles of physics, from motion to energy.	Yes	No	thegaelord	Classical Mechanics,Quantum Physics,Electromagnetism	May 4, 2025
Test Question Set #3	Learn about the fundamental concepts of biology, including cell structure and genetics.	Yes	No	thegaelord	Molecular Biology,Ecology,General Biology	May 4, 2025
Test Question Set #4	Dive into the basics of computer science, covering algorithms and data structures.	No	No	thegaelord	Data Structures and Algorithms,Artificial Intelligence,Database Systems	May 4, 2025
Test Question Set #5	Review essential English grammar rules and improve your writing skills.	Yes	No	thegaelord	English	May 4, 2025
Test Question Set #6	Get an overview of world history, from ancient civilizations to modern times.	Yes	No	thegaelord	World History,Ancient Civilizations	May 4, 2025
Test Question Set #7	Explore the key events and developments in European history from the Renaissance to present day.	Yes	No	thegaelord	Modern European History	May 4, 2025
Test Question Set #8	Challenge yourself with advanced topics in C++ programming, including data structures and algorithms.	No	No	thegaelord	Data Structures and Algorithms	May 4, 2025
Test Question Set #9	Explore the world of organic chemistry, including the structure and reactions of organic molecules.	Yes	No	thegaelord	Organic Chemistry,Inorganic Chemistry,Biochemistry	May 4, 2025

Figure 3.3.4.3 Sign out

- (1) User clicks on the button next to the account image.
- (2) User clicks on the sign out button to complete signing out.

3.3.5. View account profile

3.3.5.1. Learner view account profile

The screenshot shows a learning platform interface. At the top, there is a navigation bar with a green 'HOME' button, a 'EXPLORE' button, a search bar containing 'Search with keyword', and several other icons including 'EN' (language), a plus sign, a '100' badge, and a user profile picture. A red box labeled '2' highlights the user profile picture. A red box labeled '1' highlights the 'Profile' option in the dropdown menu that appears when the profile picture is clicked. The main content area displays two course cards: 'This title' (by nta123, 2 questions, 'Not started') and 'Test Question Set #2' (by thegaelord, 15 questions, 'Not started'). Below these cards is a quote by Benjamin Franklin: "Tell me and I forget. Teach me and I remember. Involve me and I learn." - Benjamin Franklin. At the bottom, there are two more course cards: 'Data Structures and Algorithms' (Test Question Set #4) and 'Test Question Set #8'.

Figure 3.3.5.1.a: View account profile - 1

- (1) User clicks on profile picture
- (2) User clicks on Profile button to be redirected to account page

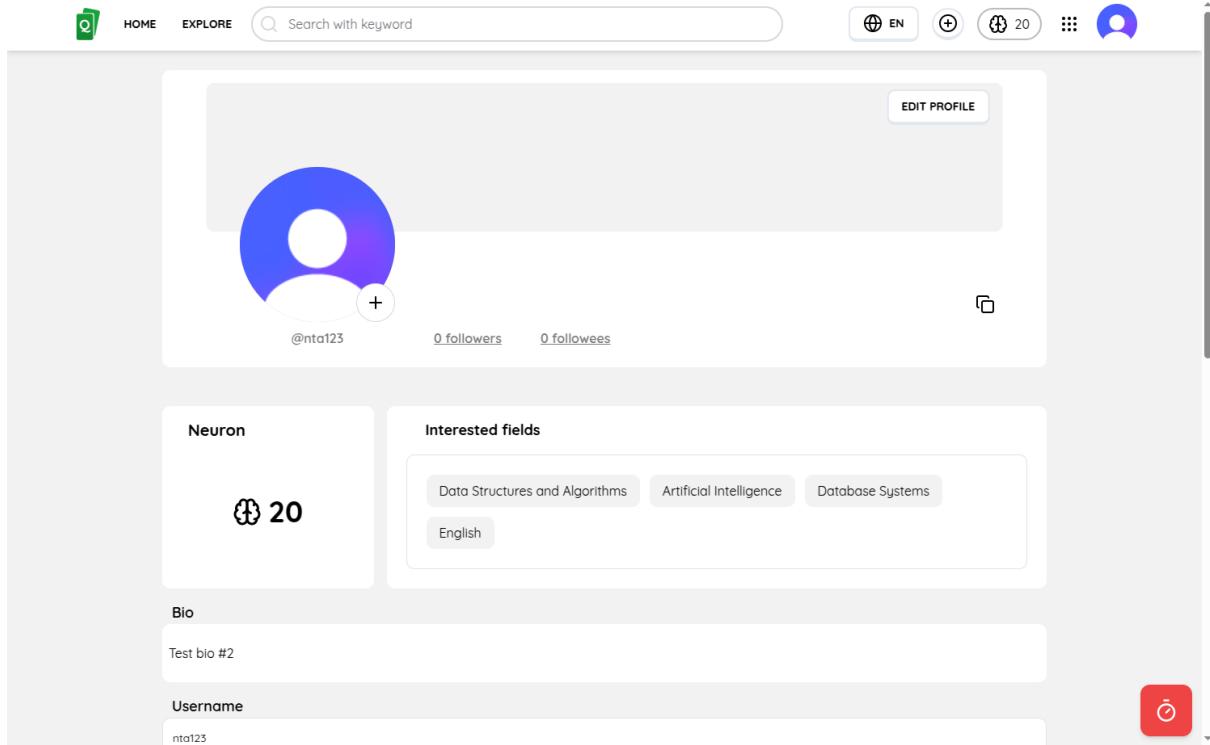


Figure 3.3.5.1.b: View account profile - 2

3.3.5.2. Administrator view account profile

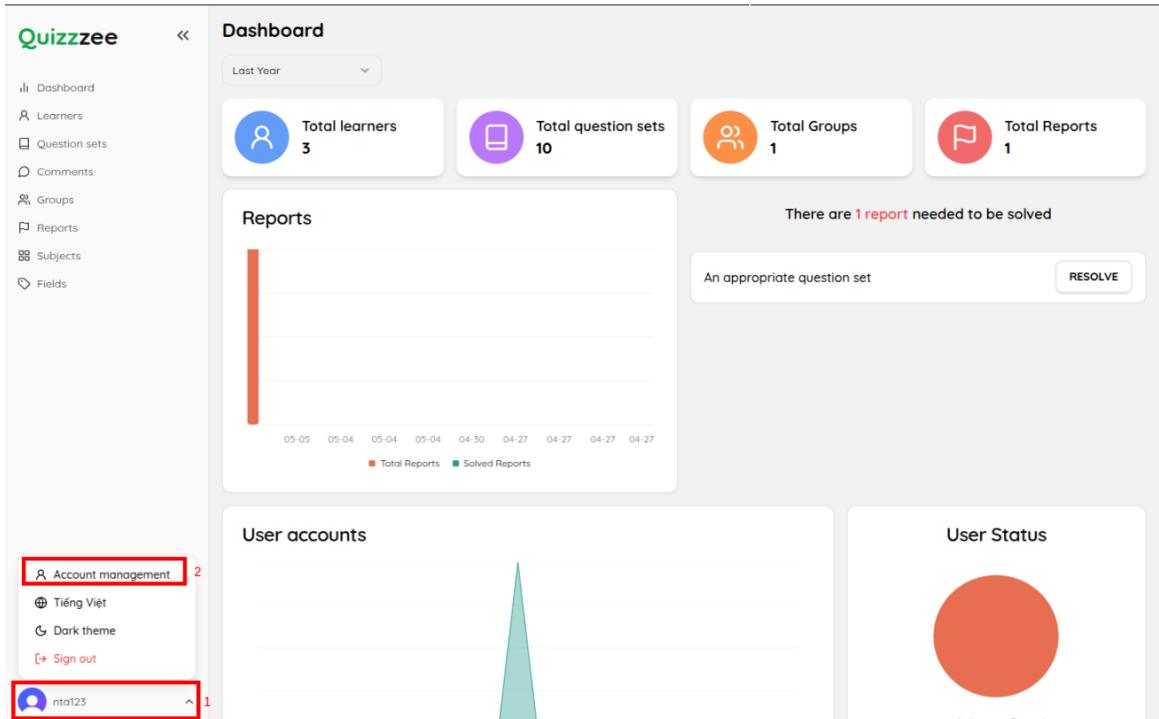


Figure 3.3.5.2.a: View account profile - 1

- (1) User clicks on profile button
- (2) User clicks on Manage account to view profile

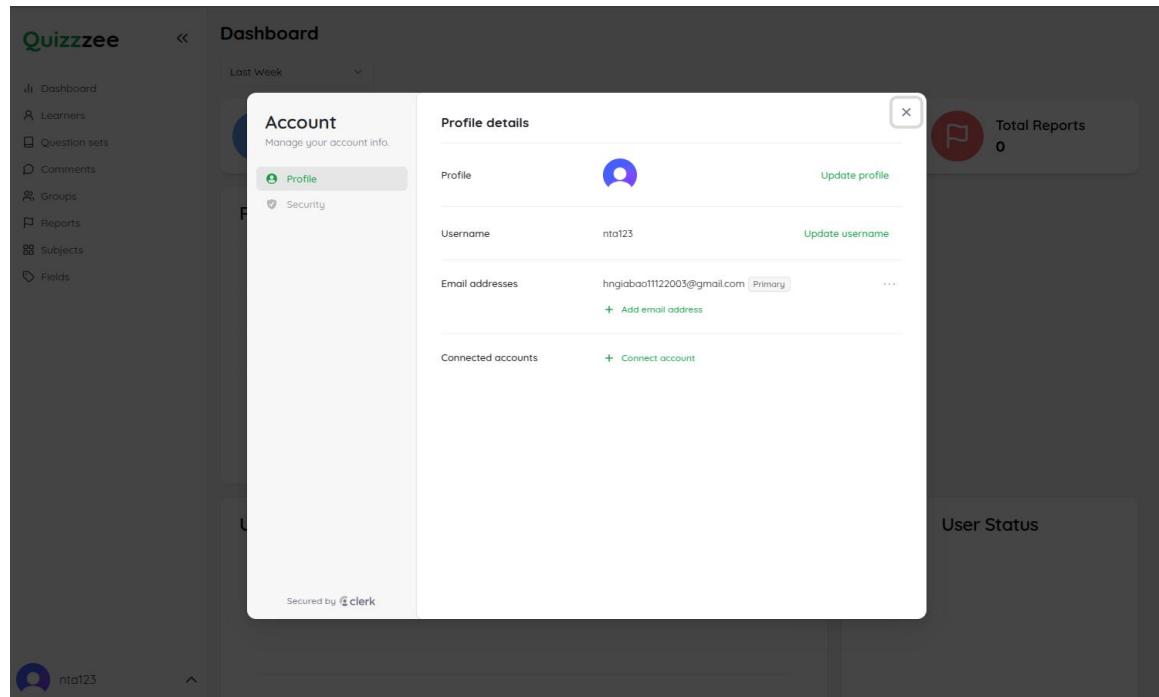


Figure 3.3.5.2.b: View account profile - 2

3.3.6. Update account profile

3.3.6.1. Learner change profile image

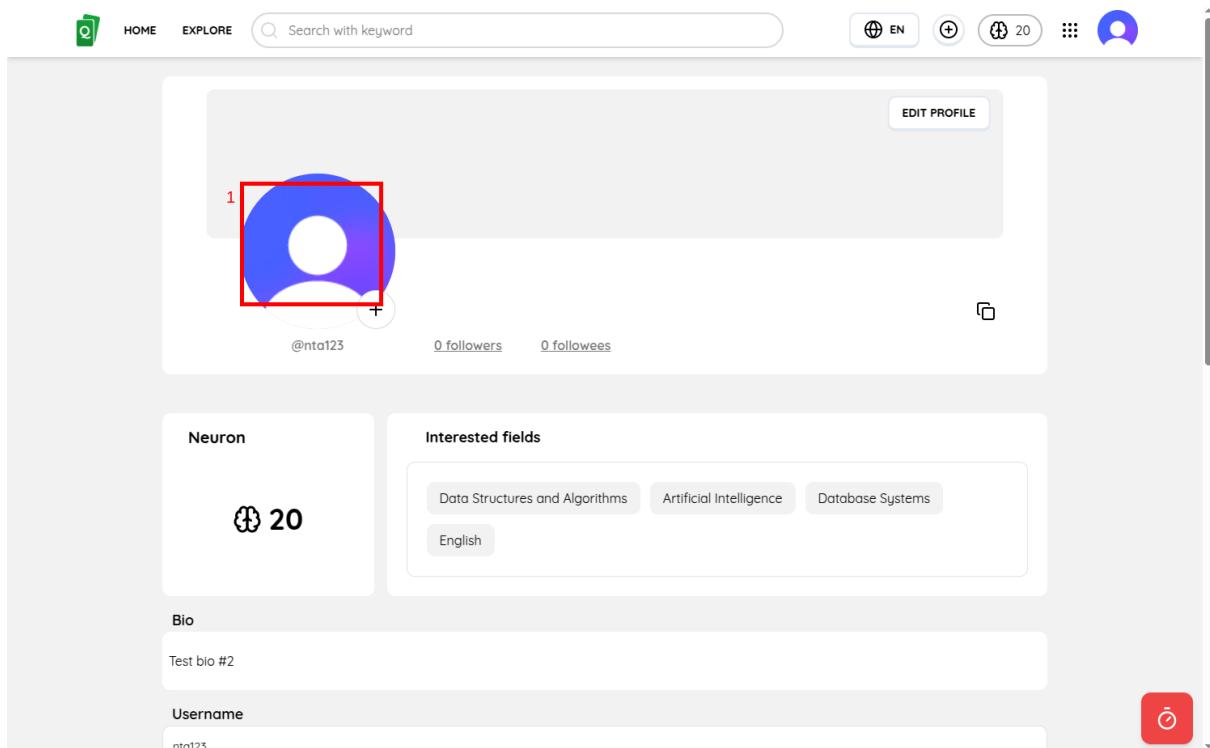


Figure 3.3.6.1.a: Change profile image - 1

- (1) Inside the account page, user clicks on profile image

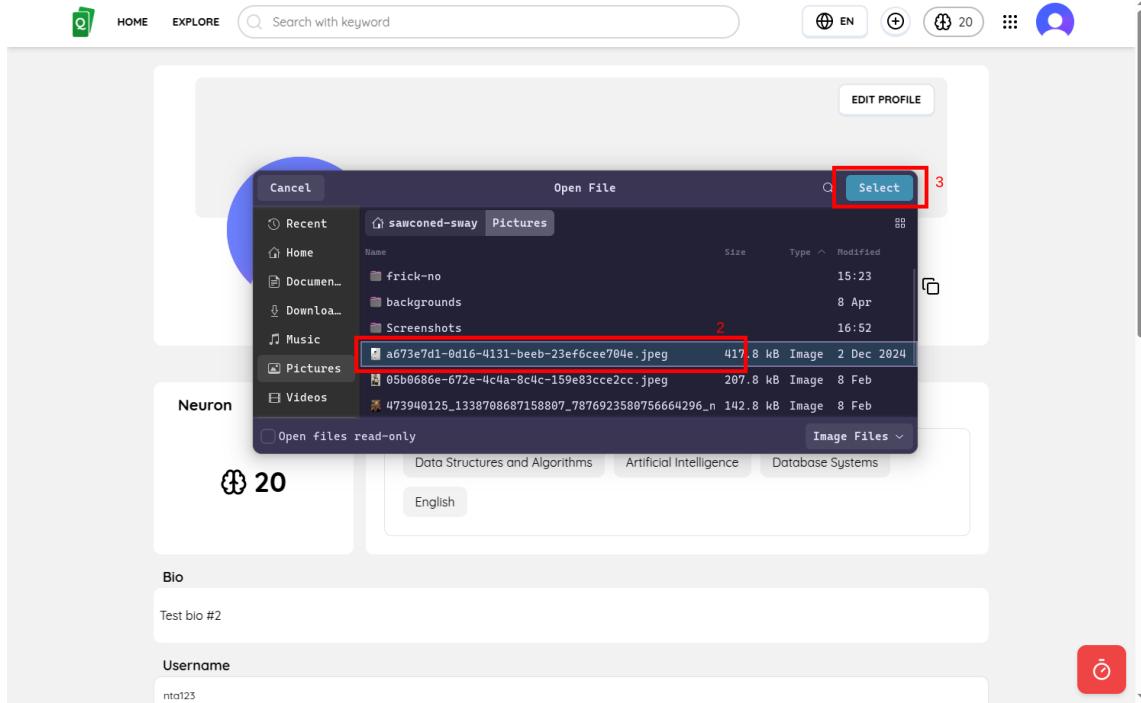


Figure 3.3.6.1.b: Change profile image - 2

- (2) User chooses an image
- (3) User clicks Select button to complete change profile image

3.3.6.2. Administrator change profile image

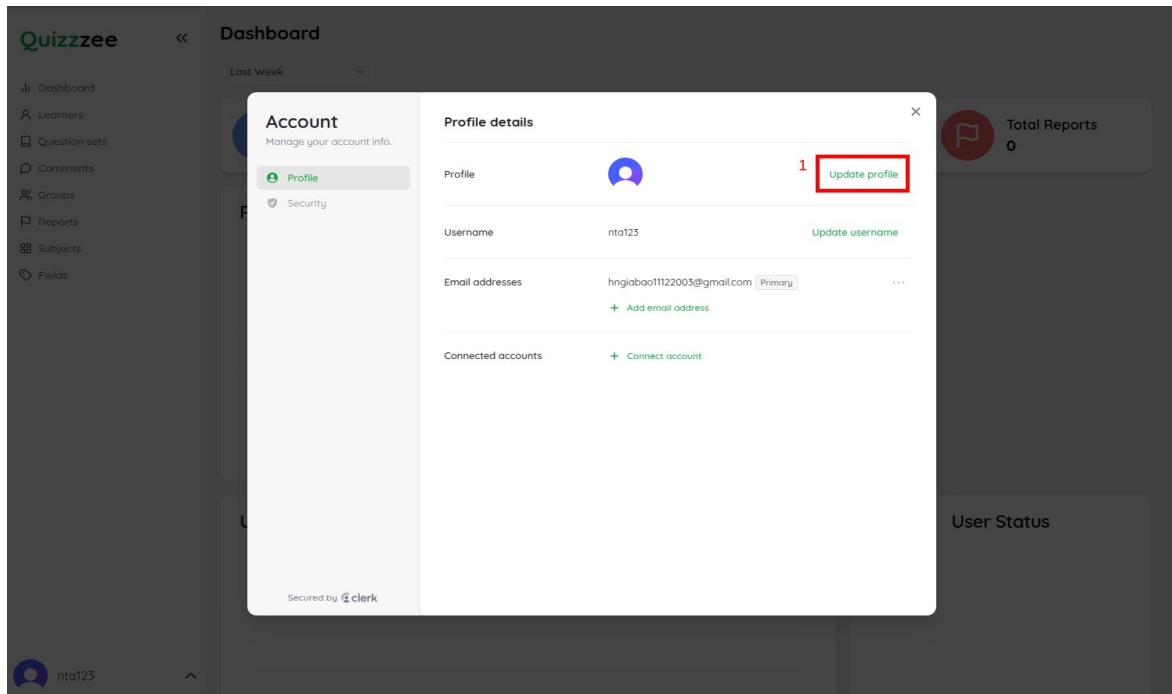


Figure 3.3.6.2.a: Change profile image - 1

- (1) From the profile popup, User clicks Update profile button

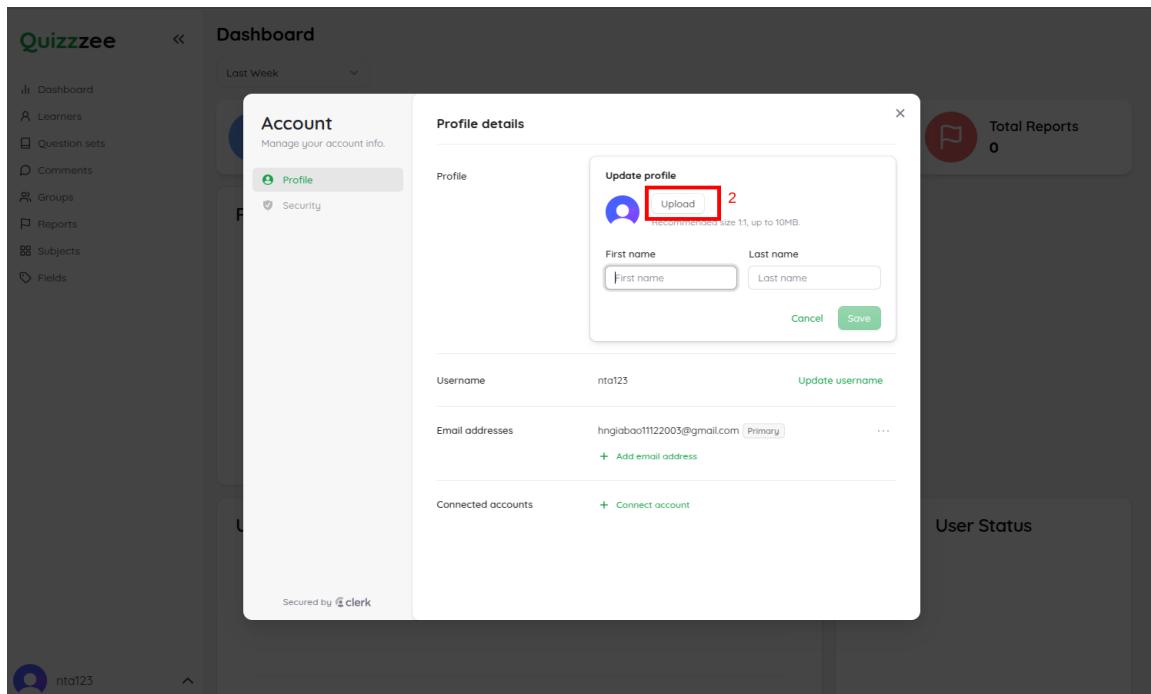


Figure 3.3.6.2.b: Change profile image - 2

- (2) User clicks Upload button

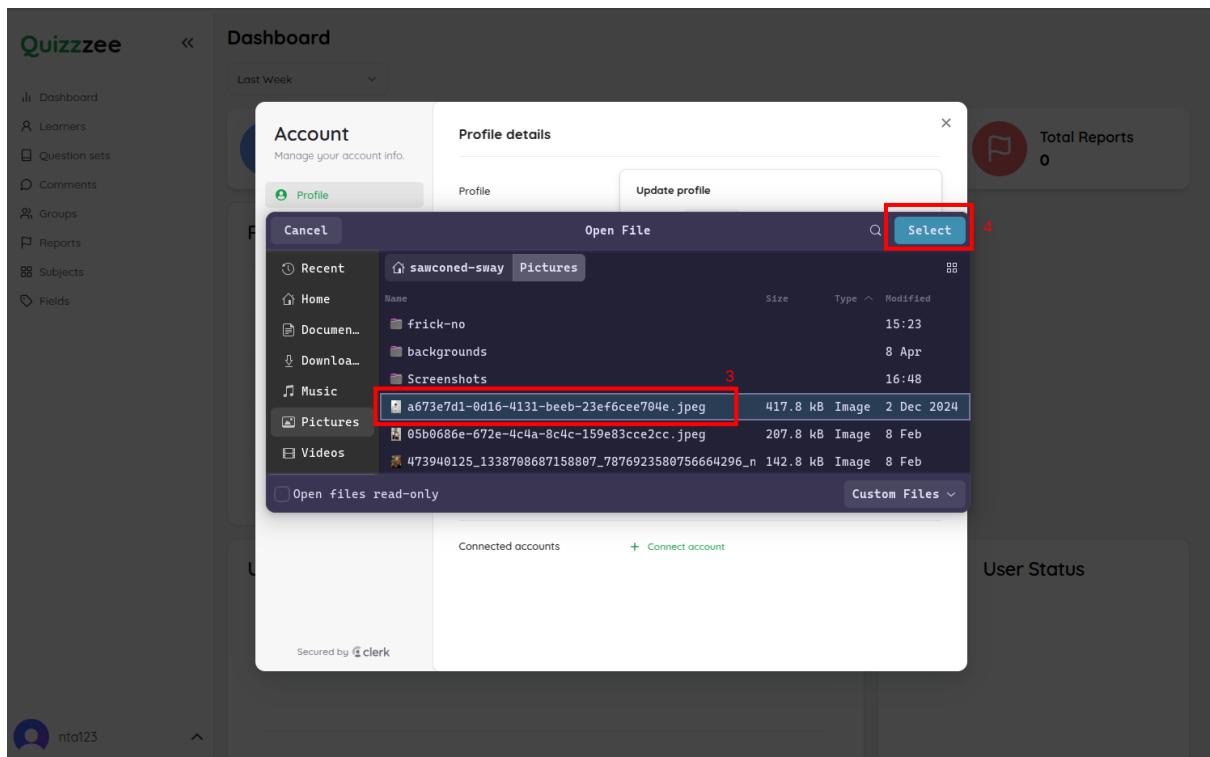


Figure 3.3.6.2.c: Change profile image - 3

- (3) User chooses an image
- (4) User clicks Select button to complete change profile image

3.3.6.3. Learner update profile information

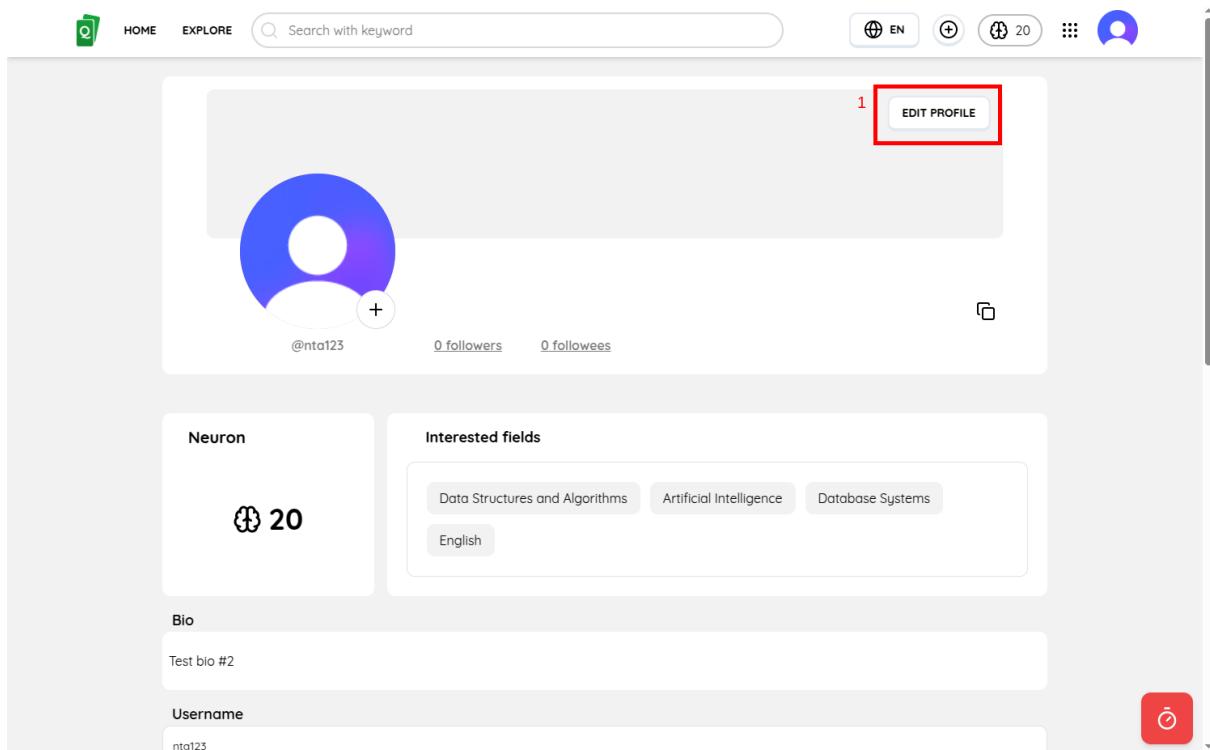


Figure 3.3.6.3.a: Update profile information - 1

- (1) User clicks Edit profile button

The screenshot shows a user profile update interface. At the top left is a circular icon with a brain-like symbol and the number '20'. To its right is a section titled 'Interested fields' containing three buttons: 'Data Structures and Algorithms', 'Artificial Intelligence', and 'Database Systems'. Below this is a 'Bio' section with a text input field containing 'Test bio #2', which is highlighted with a red box. A row of text editing icons follows. Further down are fields for 'Username' (containing 'nita123') and 'Email' (containing 'hngiabao11122005@gmail.com'). At the bottom is a 'Full name' field containing 'Dat Do Duc', also highlighted with a red box. To the right of this field is a 'SUBMIT' button, which is highlighted with a red box and has a red number '5' above it. A small circular icon with a dot is located at the bottom right.

Figure 3.3.6.3.b: Update profile information - 2

- (2) User updates field of interest (optional)
- (3) User updates bio (optional)
- (4) User updates full name (optional)
- (5) User clicks Update button to complete update profile information

3.3.6.4. Administrator update profile information

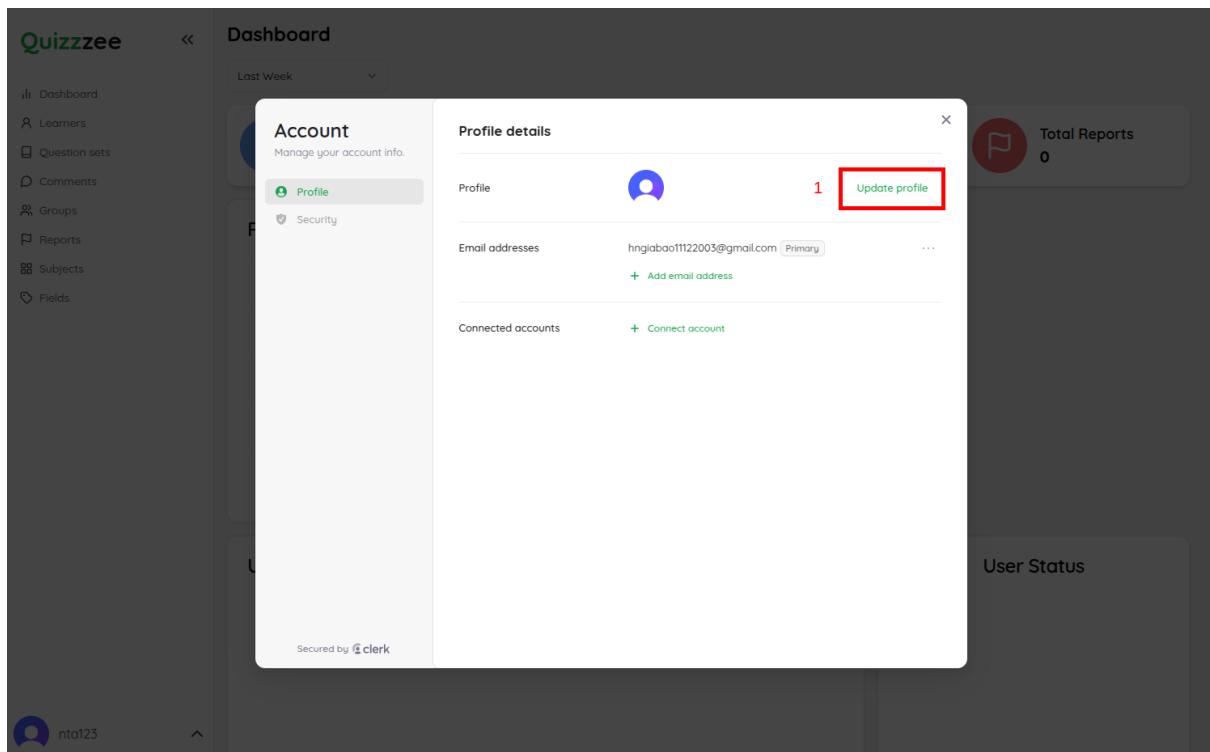


Figure 3.3.6.4.a: Update profile information - 1

- (1) Inside the account popup, user clicks Update profile button

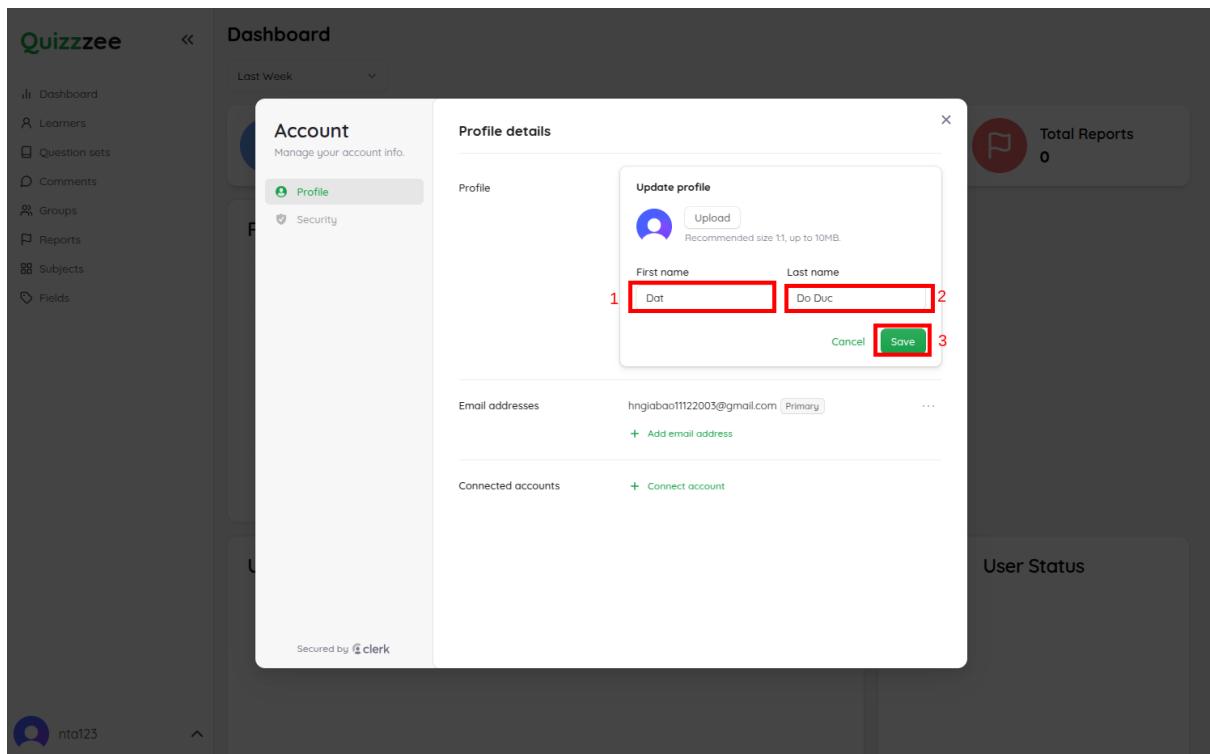


Figure 3.3.6.4.b: Update profile information - 2

- (2) User update first name(optional)
- (3) User update last name (optional)
- (4) User clicks Save to complete update profile image

3.3.6.5. Learner update status

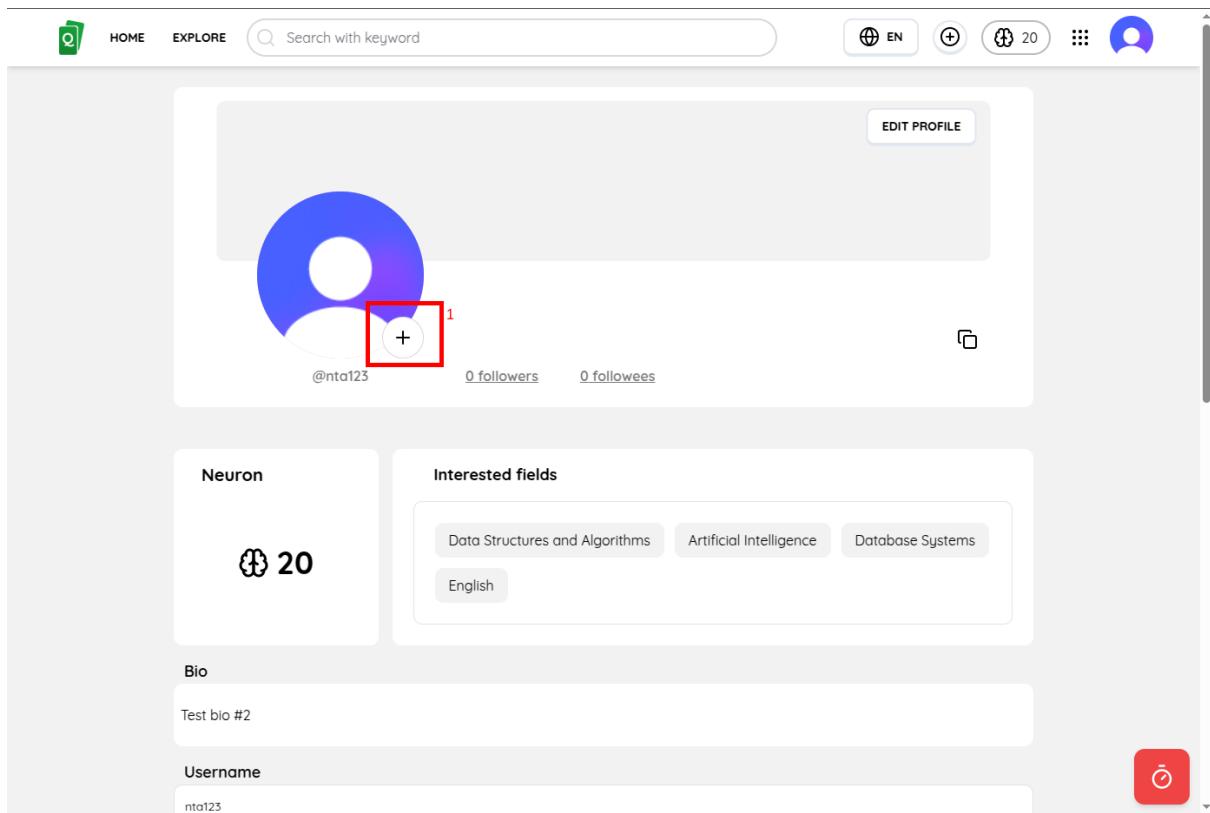


Figure 3.3.6.5.a: Update status - 1

- (1) Users click on status button

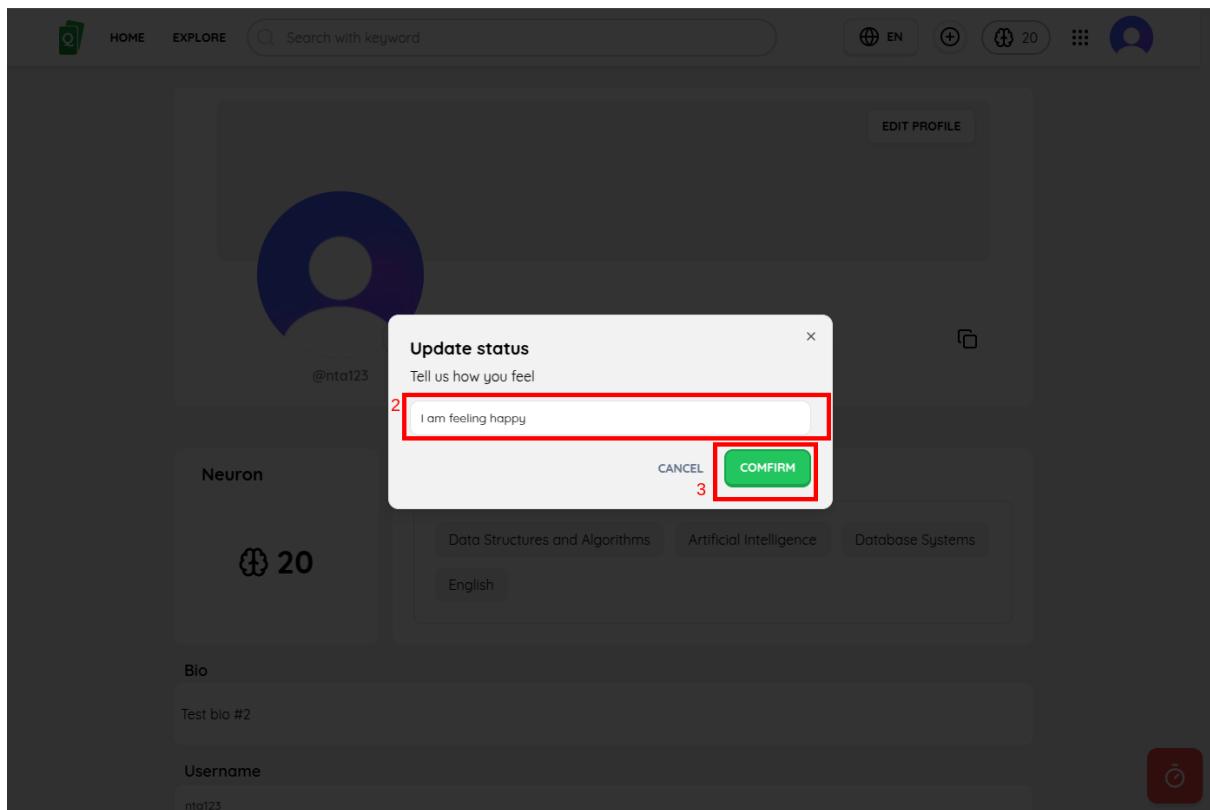
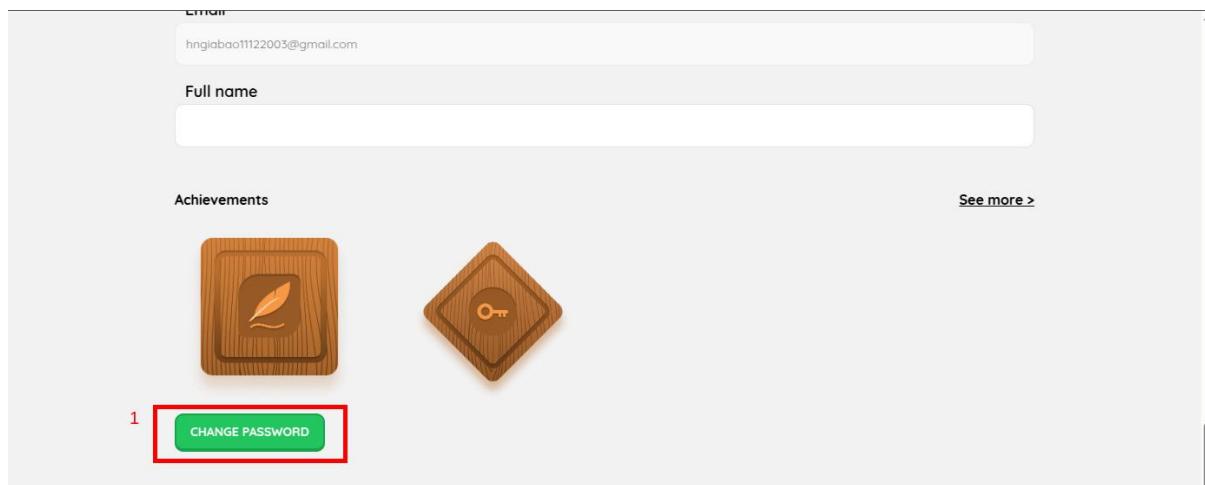


Figure 3.3.6.5.b: Update status - 2

- (2) Users inputs a status
- (3) Users clicks Confirm button

3.3.7 Update password

3.3.7.1. Learner change password



The screenshot shows the Quizzzee account settings page. At the top, there are input fields for 'Email' (containing 'hingjiaoboi1122003@gmail.com') and 'Full name'. Below these is a section titled 'Achievements' with two wooden badge icons. To the right of the achievements is a link 'See more >'. At the bottom left, there is a green button labeled 'CHANGE PASSWORD' with a red number '1' above it, indicating a user interaction point. A red rectangular box highlights this button.

Quizzzee
Helping you master knowledge,
one card at a time

Home →
Explore →
Profile →

Contact us

0123456789
quizzzee192@gmail.com

© 2025 Quizzzee. All rights reserved.

Figure 3.3.7.1.a: Change password - 1

- (1) Inside the account page, the user clicks on the edit password button

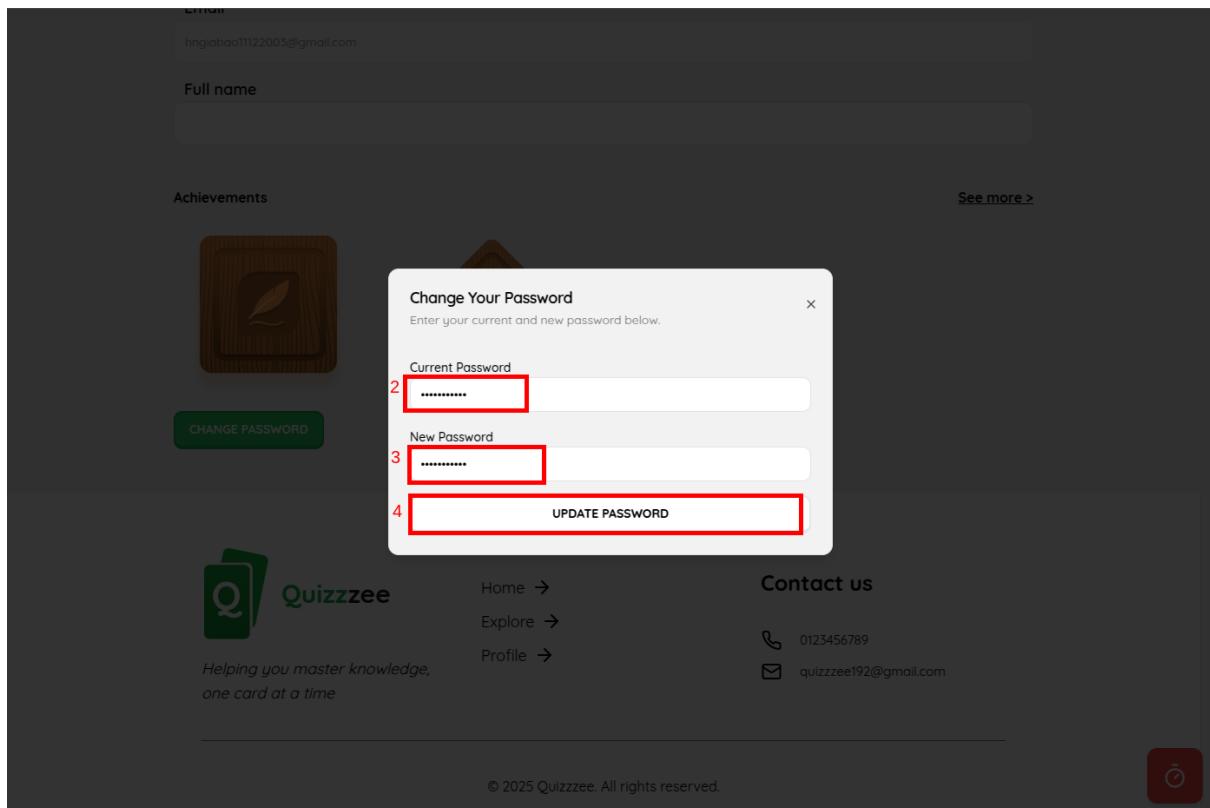


Figure 3.3.7.1.b: Change password - 2

- (2) User inputs current password
- (3) User inputs new password
- (4) User clicks on the update password button

3.3.7.2. Administrator change password

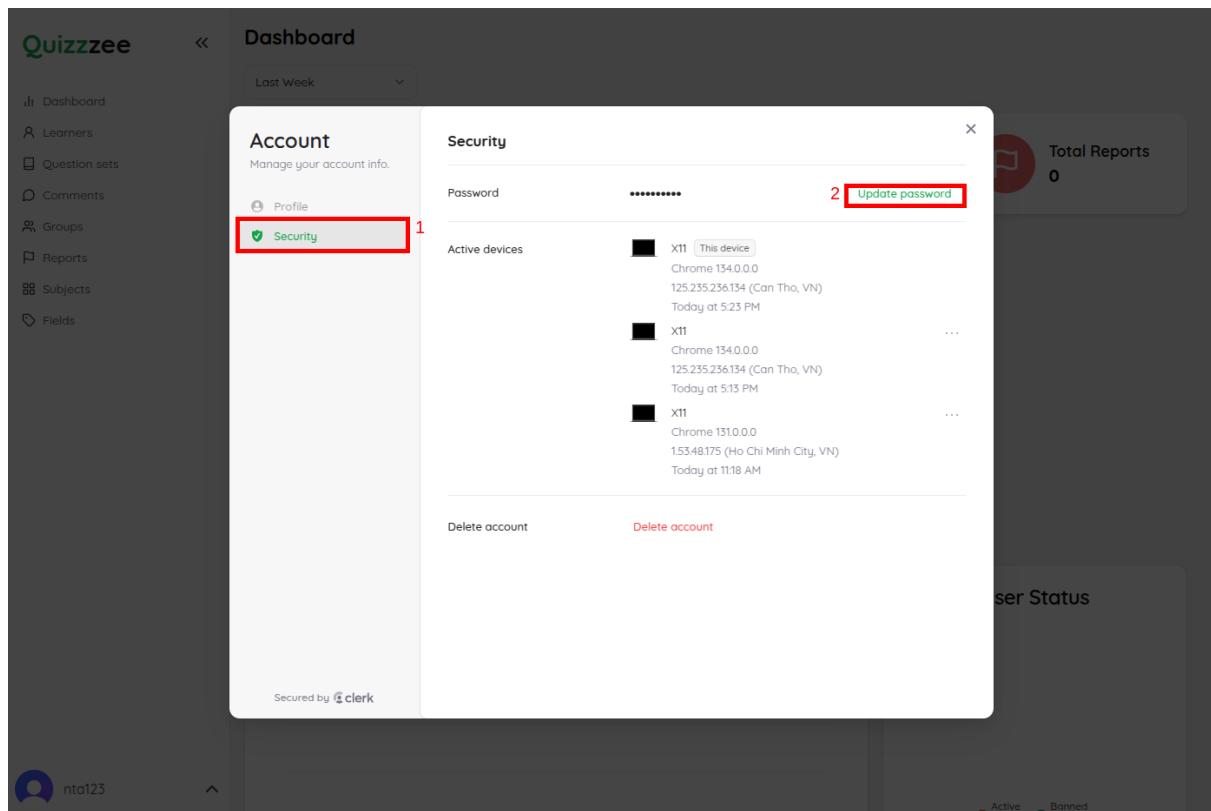


Figure 3.3.7.2.a: Change password - 1

- (1) Inside account modal, the user clicks on the security tab
- (2) User clicks on the update password button

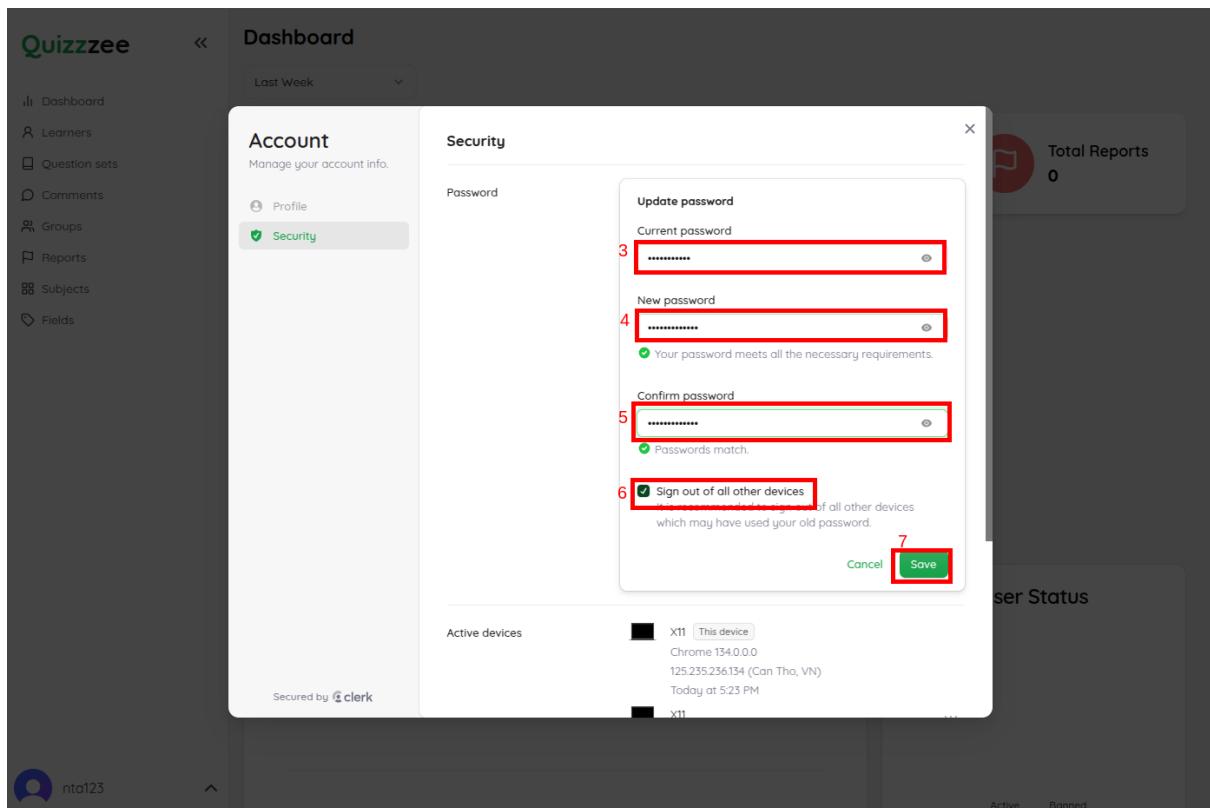


Figure 3.3.7.2.b: Change password - 2

- (3) User inputs current password
- (4) User inputs new password
- (5) User confirms password
- (6) User checks the sign out checkbox (optional)
- (7) User clicks save to complete changing password

3.3.8 Learner delete account

The screenshot shows the 'Account' section of the Quizzzee app's settings. At the top, there are links for 'Profile', 'System', and '1 Account'. Below these are sections for 'Reset settings' (with a 'RESET' button) and 'Account Settings' (with a 'UPDATE' button). Under 'Account Settings', there are buttons for 'Update settings to server', 'Sign Out' (with a 'SIGN OUT' button), and 'Delete Account' (with a 'DELETE' button, which is highlighted with a red box and has a red number '2' above it). The bottom of the screen shows the Quizzzee logo, navigation links for Home, Explore, and Profile, and contact information for the app.

© 2025 Quizzzee. All rights reserved.

Figure 3.3.8.a: Delete account - 1

- (1) Inside the settings page, the user scrolls to the delete account section.
- (2) User clicks on Delete account button

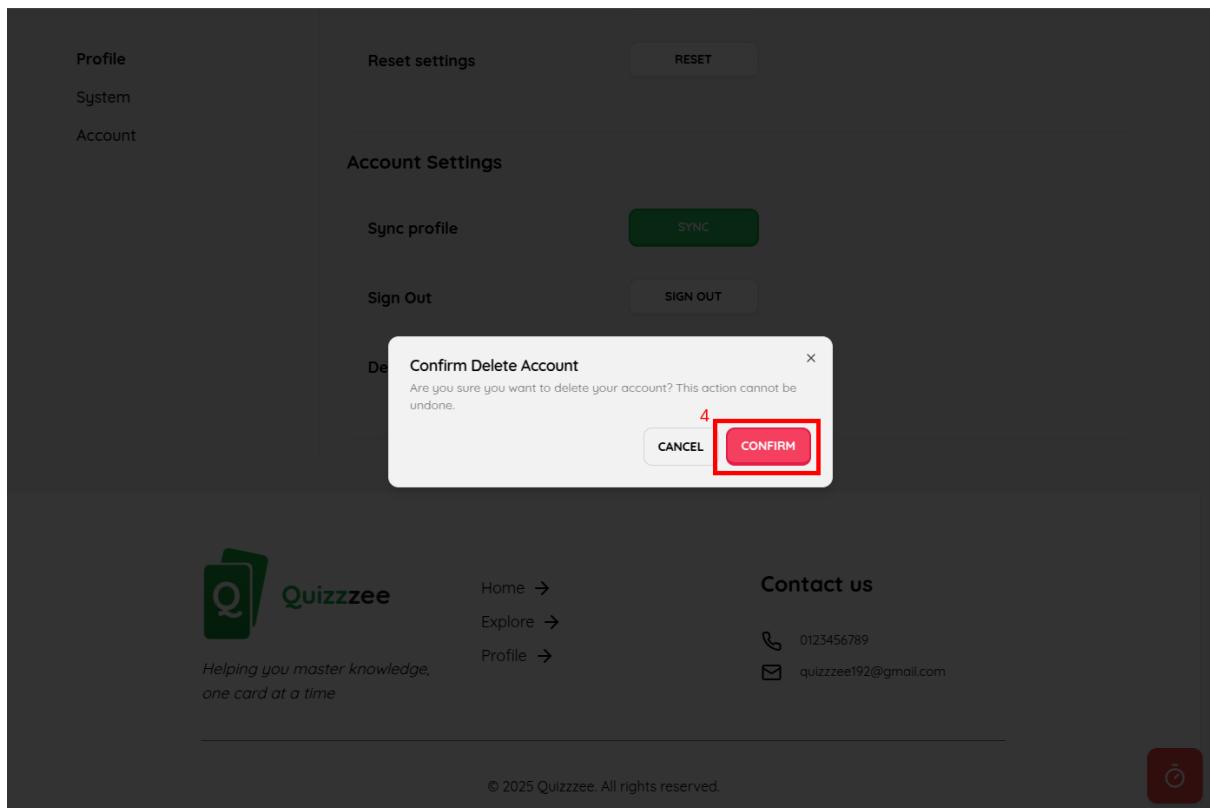


Figure 3.3.8.b: Delete account - 2

- (4) User clicks on Confirm button to complete delete account

3.3.9 Administrator and Learner reset password

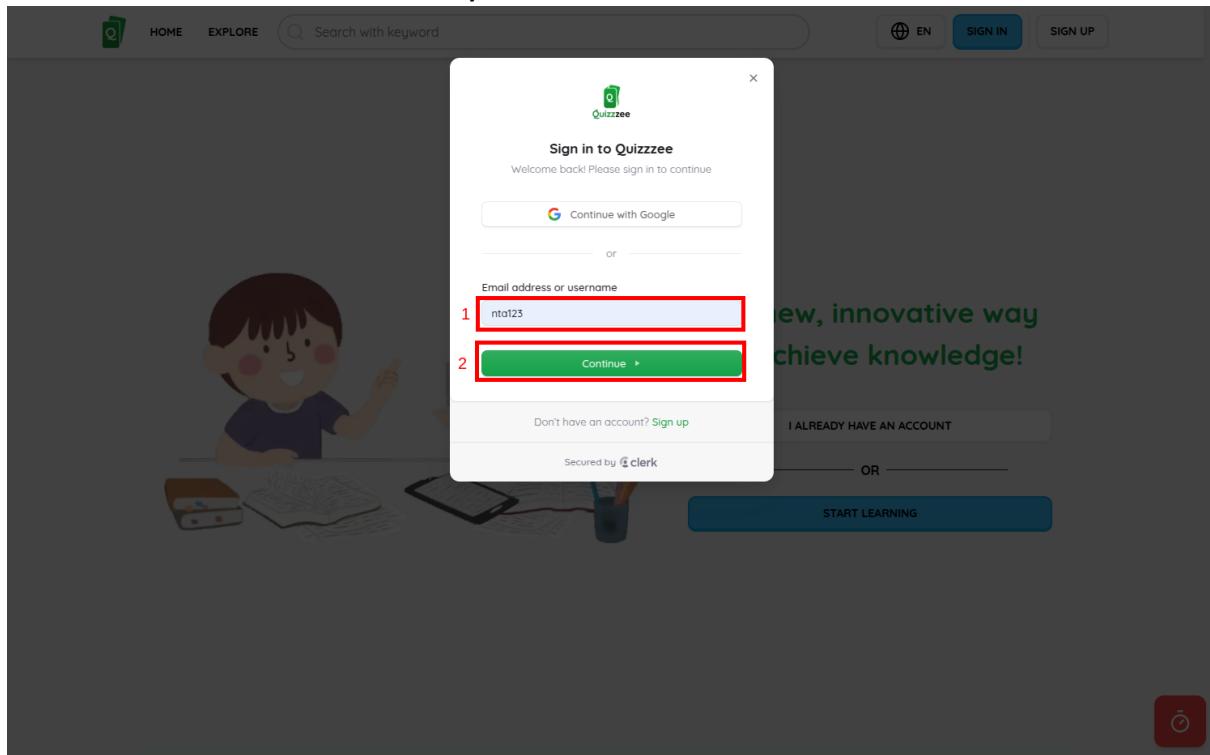


Figure 3.3.9.a: Reset password - 1

- (1) Inside the account modal, the user enters username or password
- (2) User clicks Continue button

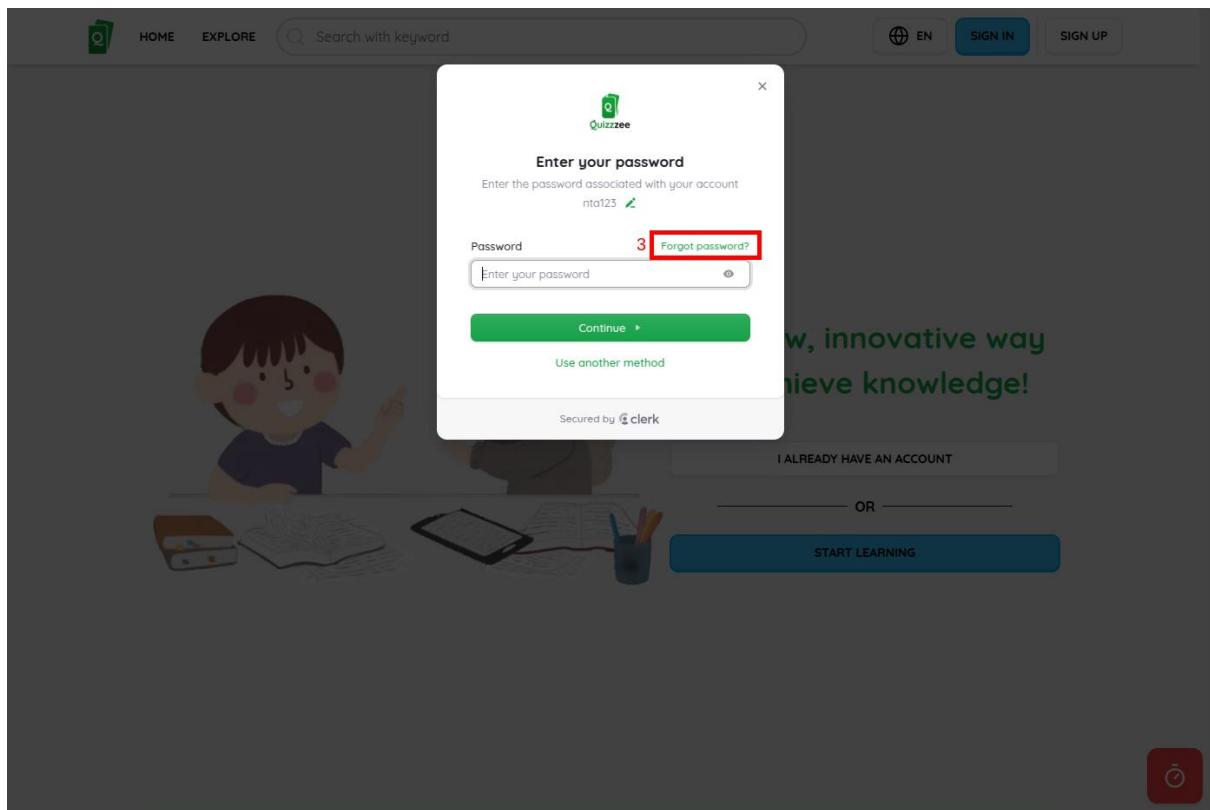


Figure 3.3.9.b: Reset password - 2

- (3) User clicks forgot password? button

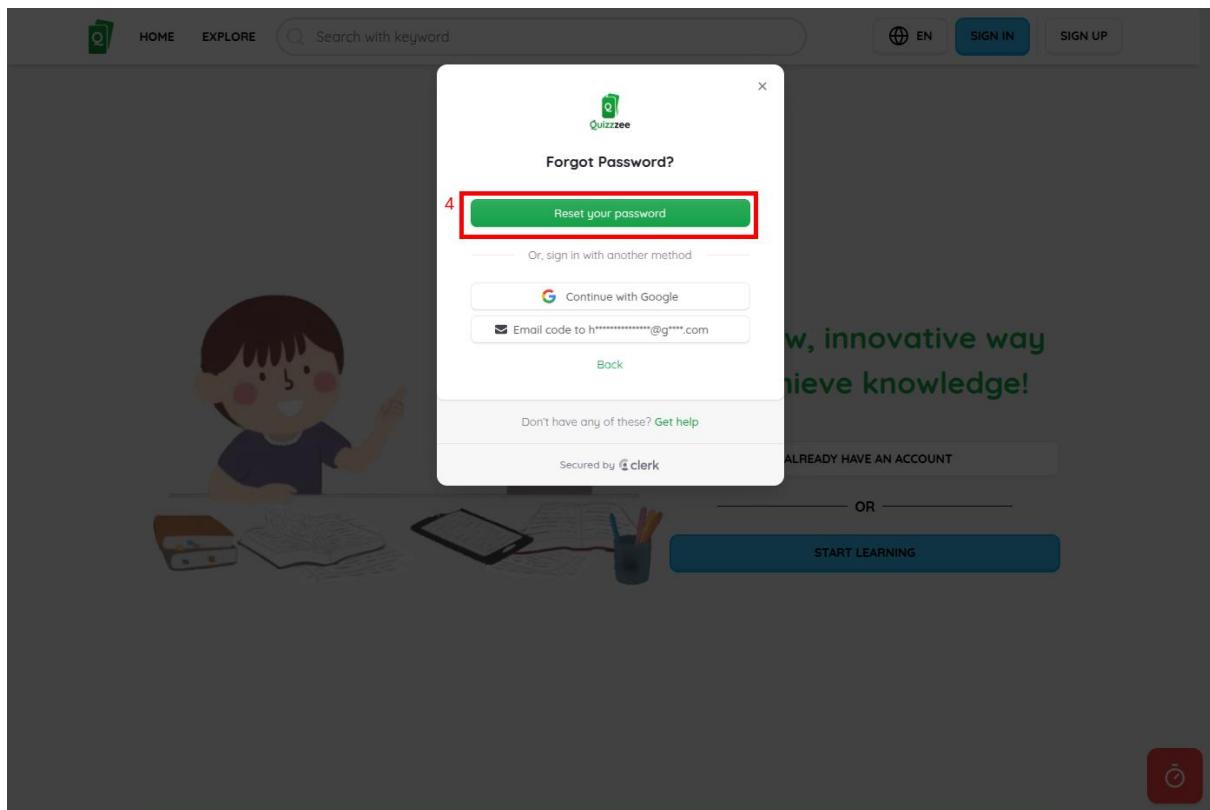


Figure 3.3.9.c: Reset password - 3

- (4) User clicks reset your password button

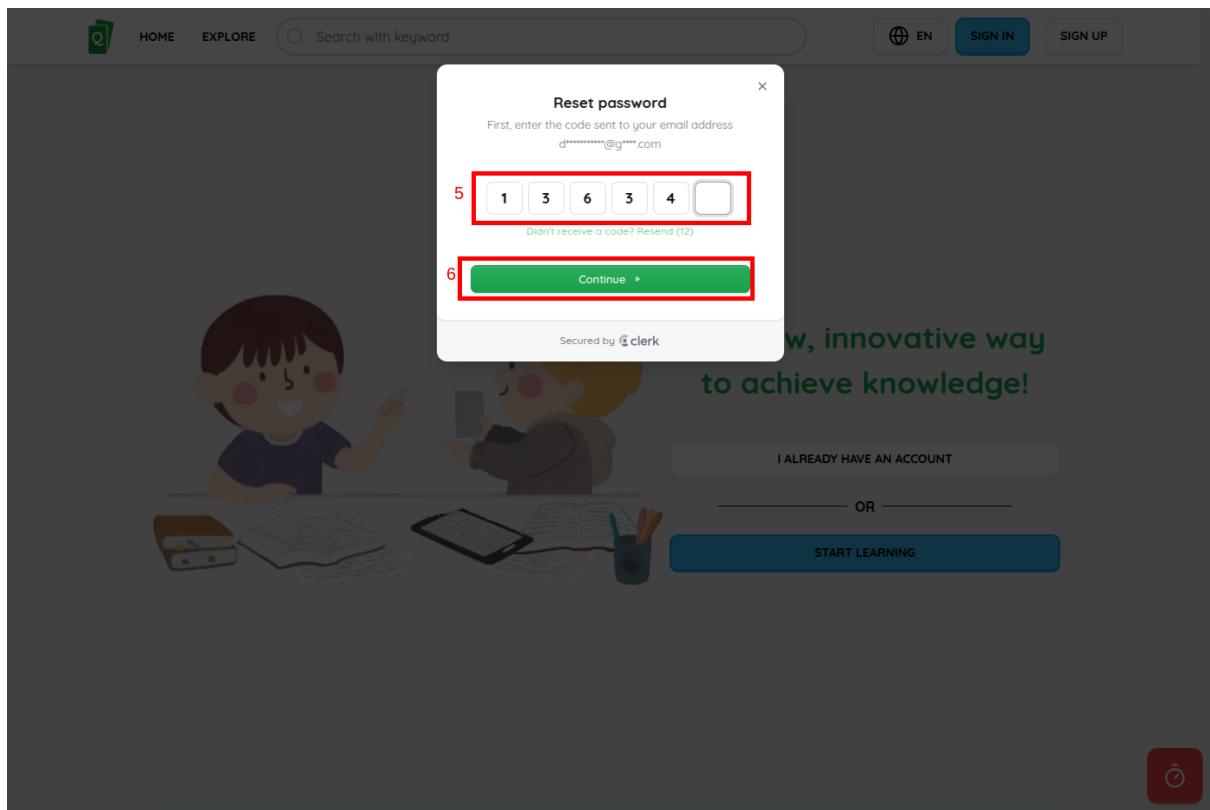


Figure 3.3.9.d: Reset password - 4

- (5) User input code
- (6) User clicks Continue button

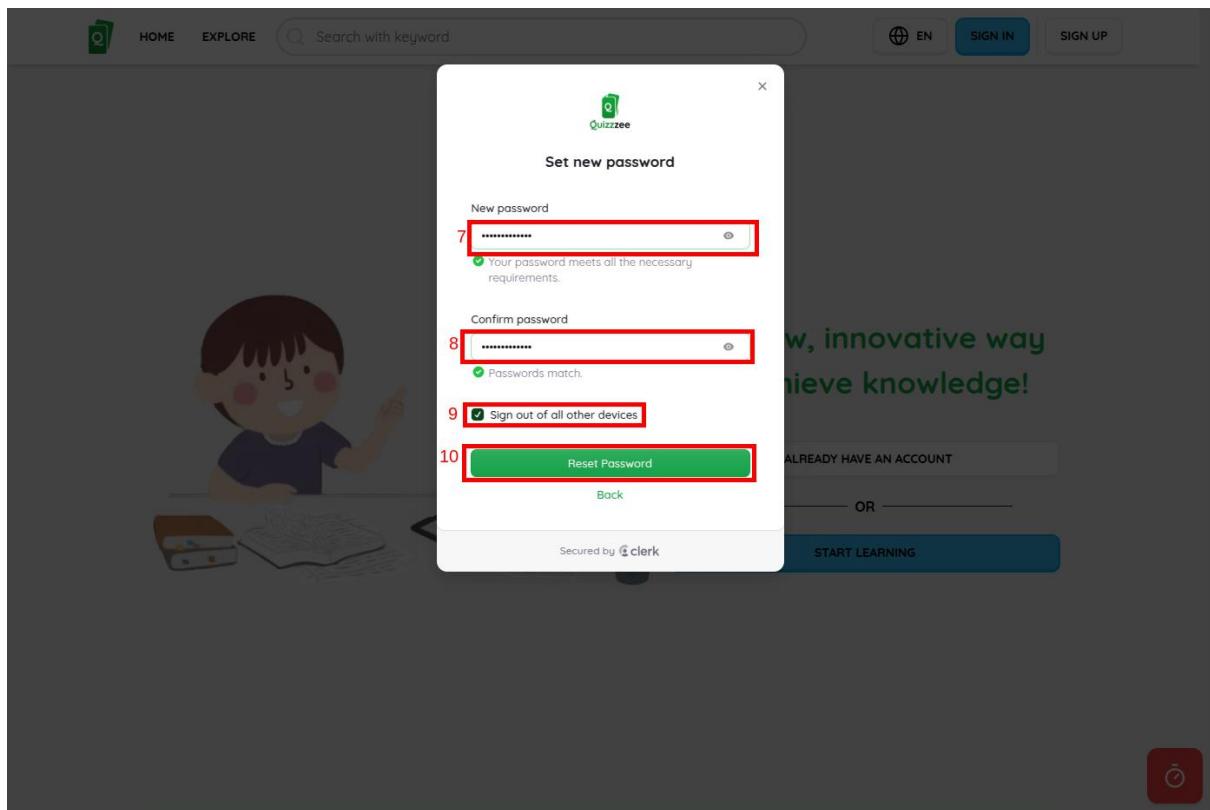


Figure 3.3.9.e: Reset password - 5

- (7) User enters new password
- (8) User confirms new password
- (9) User clicks sign out of all device checkbox (optional)
- (10) User clicks Reset password button to complete reset password

3.3.10 View question sets

3.3.10.1 Learner and Guest view from home page

The screenshot shows a user interface for a learning platform. At the top, there is a navigation bar with icons for HOME, EXPLORE, and a search bar labeled "Search with keyword". To the right of the search bar are buttons for language (EN), notifications (20), and user profile.

The main content area displays three question sets:

- A title**: Created by nta123, 2 questions. Description: A description. Categories: General Biology. Rating: 0.
- This title**: Created by nta123, 2 questions. Description: this description. Categories: General Biology, Electromagnetism. Rating: 0.
- Test Question Set #2**: Created by thegaelord, 15 questions. Description: Understand the basic principles of physics, from motion to energy. Categories: Classical Mechanics, Quantum Physics, Electromagnetism. Rating: 4.0.

At the bottom of the page, there is a quote: "A wise man can learn more from a foolish question than a fool can learn from a wise answer." To the right of the quote is a red circular button with a white icon.

Figure 3.3.10.1: View question sets

- (1) Users goes to home page to view question sets

3.3.10.2 Learner view from created / favorited sets

The screenshot shows a user's dashboard on a learning platform. At the top, there are navigation links for 'HOME' and 'EXPLORE', a search bar, and a language selector set to 'EN'. A 'More' button (indicated by a red box with '1') is highlighted, and a dropdown menu appears with options: 'My Groups' (with a red box and '2'), 'My Sets' (highlighted with a red box), 'Goals', and 'Achievements'. Below the header, a section titled 'Pick up where you left' contains two card-like items:

- Test Question Set #2**: Created by 'thegaelord' (15 questions). Description: Understand the basic principles of physics, from motion to energy. Tags: Classical Mechanics, Quantum Physics, Electromagnetism. Rating: 4.0 stars.
- This title**: Created by 'nta123' (2 questions). Description: this description. Tags: General Biology, Electromagnetism. Rating: 0 stars.

In the center of the page is a quote box with the following text:
"Live as if you were to die tomorrow. Learn as if you
were to live forever."
- Mahatma Gandhi

At the bottom, there is a section titled 'Data Structures and Algorithms' with two cards:

- Test Question Set #4**
- Test Question Set #8**

Figure 3.3.10.2.a: View question sets - 1

- (1) User clicks on More button
- (2) User clicks on My Sets button to be redirected to Sets page

The screenshot shows a user interface for managing question sets. At the top, there are navigation links for 'HOME' and 'EXPLORE', a search bar labeled 'Search with keyword', and various user account and system status icons. Below this, the title 'Your Question Sets' is displayed, followed by the subtitle 'Manage and organize your created and favorite question sets.' A navigation bar includes tabs for 'Created' (which is selected) and 'Favorite', along with a search input field and a 'CREATE' button.

The main content area displays a single question set card. The card has a title 'This title', a creation timestamp '@ @nta123', a question count '2 questions', and a description 'this description'. It features two category tags: 'Electromagnetism' (blue) and 'General Biology' (yellow). Below the tags is a five-star rating icon. The URL 'localhost:3000/set-detail/6817282ea50082780abd9f8a' is visible at the bottom of the card.

On the right side of the page, there is a vertical sidebar with a red circular icon containing a white question mark symbol. The page footer indicates '1/1'.

Figure 3.3.10.2.b: View question sets - 2

- (3) User views created / favorited question sets from Sets page

3.3.10.3 The Administrator views question sets inside the question sets page.

The screenshot shows the Quizzzee application interface. On the left, there is a sidebar with various navigation options: Dashboard, Learners, Question sets (which is highlighted with a red box and a red number '1'), Comments, Groups, Reports, Subjects, and Fields. Below the sidebar is a user profile icon for 'nto123'. The main content area is titled 'Question Sets' and displays a table with 10 rows, each representing a 'Test Question Set'. The columns in the table are: TITLE, Description, Public, Banned, Author, Subject, and Date. The first few rows of data are as follows:

TITLE	Description	Public	Banned	Author	Subject	Date
Test Question Set #1	Explore fundamental concepts in mathematics, including algebra and geometry.	Yes	No	thegaelord	Calculus,Linear Algebra,Statistics	May 4, 2025
Test Question Set #2	Understand the basic principles of physics, from motion to energy.	Yes	No	thegaelord	Classical Mechanics,Quantum Physics,Electromagnetism	May 4, 2025
Test Question Set #3	Learn about the fundamental concepts of biology, including cell structure and genetics.	Yes	No	thegaelord	Molecular Biology,Ecology,General Biology	May 4, 2025
Test Question Set #4	Dive into the basics of computer science, covering algorithms and data structures.	No	No	thegaelord	Data Structures and Algorithms,Artificial Intelligence,Database Systems	May 4, 2025
Test Question Set #5	Review essential English grammar rules and improve your writing skills.	Yes	No	thegaelord	English	May 4, 2025
Test Question Set #6	Get an overview of world history, from ancient civilizations to modern times.	Yes	No	thegaelord	World History,Ancient Civilizations	May 4, 2025
Test Question Set #7	Explore the key events and developments in European history from the Renaissance to present day.	Yes	No	thegaelord	Modern European History	May 4, 2025
Test Question Set ...	Challenge yourself with advanced topics in C++ programming, including	No	No	thegaelord	Data Structures and Algorithms	May 4, 2025

Figure 3.3.10.3 View question sets

- (1) User clicks on the question sets button to be redirected to the question sets page.
- Inside the question sets page, the user can see the table of learners.

3.3.11 View question set detail

3.3.11.1 Learner and guest view question set detail

The screenshot shows a user interface for a learning platform. At the top, there is a navigation bar with icons for HOME, EXPLORE, and a search bar labeled "Search with keyword". On the right side of the top bar are buttons for language (EN), notifications (20), and user profile.

The main content area displays a list of question sets. The first item in the list is highlighted with a red border. This item is titled "Test Question Set #2" and is associated with the user "thegaelord" (15 questions). It has a brief description: "Understand the basic principles of physics, from motion to energy." Below the title are three categories: "Classical Mechanics", "Quantum Physics", and "Electromagnetism". It also shows a rating of 4.0 stars. The number "1" is positioned above the first item, indicating it is the current selection.

The second item in the list is titled "This title" and is associated with the user "nto123" (2 questions). It has a brief description: "this description". Below the title are two categories: "General Biology" and "Electromagnetism". It shows a rating of 0 stars. The number "1" is positioned above the first item, indicating it is the current selection.

Below the list of question sets, there is a quote in a box: "Tell me and I forget. Teach me and I remember. Involve me and I learn." - Benjamin Franklin.

At the bottom of the screen, there are two more question set cards: "Test Question Set #4" and "Test Question Set #8". A red circular button with a white play icon is located on the right side of the bottom row.

Figure 3.3.11.a: View question set detail - 1

- (1) User clicks on the question set card to be redirected to the question set detail page.

The screenshot shows a user interface for viewing a question set detail. At the top, there is a navigation bar with icons for HOME, EXPLORE, and a search bar labeled "Search with keyword". On the right side of the top bar are buttons for EN, +, a notification icon showing 20 notifications, and a user profile icon.

The main content area starts with a title "This title" and a description "About this question set: this description". Below this, it shows "Last update: 2025-05-04", "Rating: ★ ★ ★ ★ 0.0", and "Tags: General Biology, Electromagnetism". To the right of these details are several small circular icons with symbols: a download arrow, a gear, a magnifying glass, a refresh, and a star.

Below the title, there is a user profile section for "nta123" with a blue profile picture and a "FOLLOW" button. To the right of this is another user profile picture with a blue background.

The central part of the screen features a rating section with the text "I think this Quizzzee's rating should be..." followed by five empty star icons. Below this is a "Learning Progress" section containing a large orange circle representing the progress status. A legend below the circle indicates: Not Learned (orange), Learning (green), Almost done (blue), and Mastered (yellow). To the right of the progress circle are four buttons: FLASH CARD, PRACTICE, QUIZ, and PLAY.

Figure 3.3.11.b: View question set detail - 2

- (2) User view detail of question set.

3.3.11.2 The Administrators view question sets inside the question sets page.

The screenshot shows the Quizzzee application interface. On the left is a sidebar with navigation links: Dashboard, Learners, Question sets (which is the active page, indicated by a red number '1'), Comments, Groups, Reports, Subjects, and Fields. The main area is titled 'Question Sets' and displays a table of 10 sets found. A search bar labeled 'Filter title...' is at the top of the table. The table has columns: TITLE, Description, Public, Banned, Author, Subject, and Date. The first row, 'Test Question Set #1', is highlighted with a red border. The details for 'Test Question Set #1' are: Title 'Test Question Set #1', Description 'Explore fundamental concepts in mathematics, including algebra and geometry.', Status 'Public', Banned 'No', Author 'thegaelord', Subject 'Calculus,Linear Algebra,Statistics', and Date 'May 4, 2025'. The other rows follow a similar structure with varying descriptions and subjects.

TITLE	Description	Public	Banned	Author	Subject	Date
Test Question Set #1	Explore fundamental concepts in mathematics, including algebra and geometry.	Yes	No	thegaelord	Calculus,Linear Algebra,Statistics	May 4, 2025
Test Question Set #2	Understand the basic principles of physics, from motion to energy.	Yes	No	thegaelord	Classical Mechanics,Quantum Physics,Electromagnetism	May 4, 2025
Test Question Set #3	Learn about the fundamental concepts of biology, including cell structure and genetics.	Yes	No	thegaelord	Molecular Biology,Ecology,General Biology	May 4, 2025
Test Question Set #4	Dive into the basics of computer science, covering algorithms and data structures.	No	No	thegaelord	Data Structures and Algorithms,Artificial Intelligence,Database Systems	May 4, 2025
Test Question Set #5	Review essential English grammar rules and improve your writing skills.	Yes	No	thegaelord	English	May 4, 2025
Test Question Set #6	Get an overview of world history, from ancient civilizations to modern times.	Yes	No	thegaelord	World History,Ancient Civilizations	May 4, 2025
Test Question Set #7	Explore the key events and developments in European history from the Renaissance to present day.	Yes	No	thegaelord	Modern European History	May 4, 2025
Test Question Set ...	Challenge yourself with advanced topics in C++ programming, including	No	No	thegaelord	Data Structures and Algorithms	May 4, 2025

Figure 3.3.11.2.a View question set detail - 1

- (1) User clicks the question set table row to be redirected to the learner detail page.

The screenshot shows the Quizzzee platform interface. On the left, there's a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. A user profile icon for 'nta123' is also present. The main content area is titled 'Test Question Set #1' and describes it as exploring fundamental concepts in mathematics, including algebra and geometry. It includes sections for Author ('thegaelord'), Rating, Subjects (Calculus / Giải tích, Linear Algebra / Đại số tuyến tính, Statistics / Thống kê), and Questions (21). Three questions are listed:

- What is the value of π (pi) to two decimal places?**
Type: multiple_choice
Correct answer: 3.14
Options: 3.15, 3.13, 3.16
- What is the square root of 81?**
Type: multiple_choice
Correct answer: 9
Options: 8, 7, 6
- What is the result of 7×8 ?**
Type: multiple_choice
Correct answer: 56
Options: 54

Figure 3.3.11.2.b View question set detail - 2

- (2) User views detail of question set.

3.3.12 Learner creates question sets

The screenshot shows a user interface for a learning platform. At the top, there is a navigation bar with 'HOME', 'EXPLORE', a search bar, and various user icons. A red box highlights the 'Create set' button in the top right corner of the header. Below the header, a section titled 'Pick up where you left' contains two items: 'This title' and 'Test Question Set #2'. Each item has a progress bar (0%), a user icon, a question count, a description, categories, and a rating. A central quote by Benjamin Franklin is displayed: "Tell me and I forget. Teach me and I remember. Involve me and I learn." - Benjamin Franklin. At the bottom, there are two tabs labeled 'Test Question Set #4' and 'Test Question Set #8', followed by a red circular icon.

Figure 3.3.12.a: Create question sets - 1

- (1) User clicks on the create icon.
- (2) User clicks on the Create set button to be redirected to the create question set page.

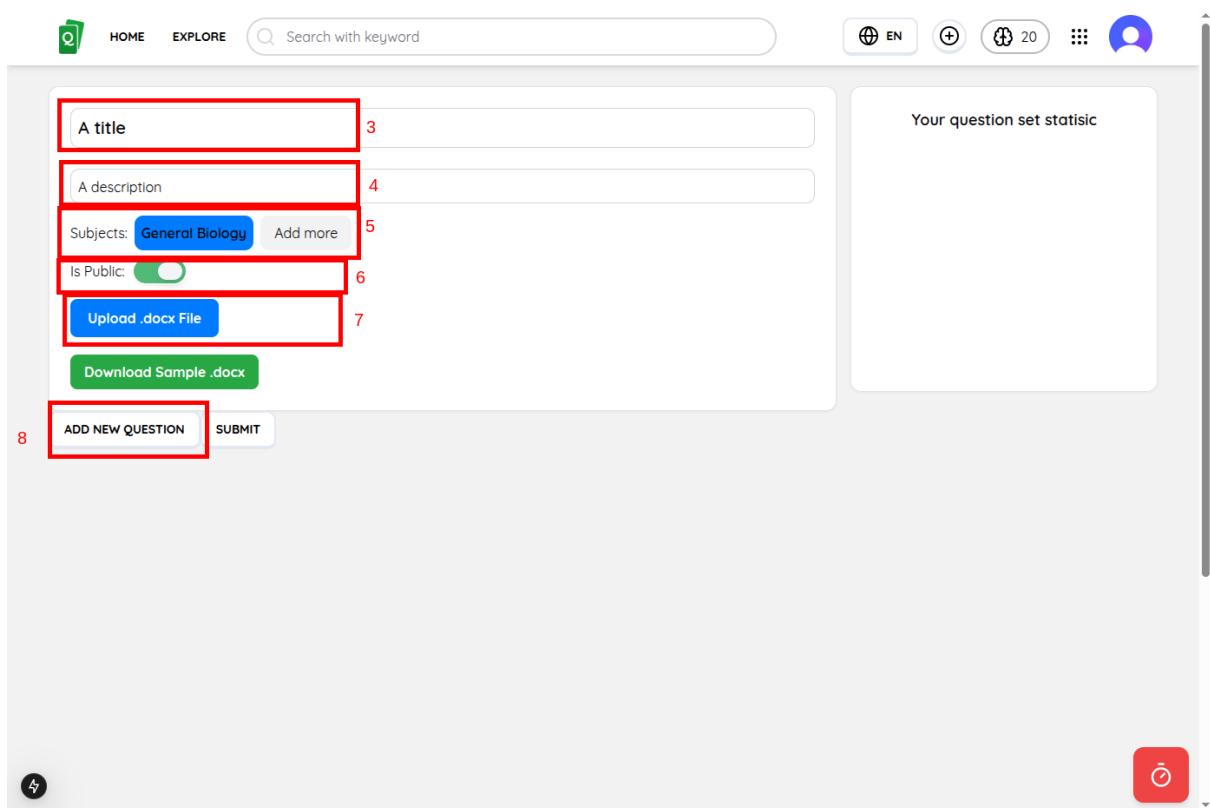


Figure 3.3.12.b: Create question sets - 2

- (3) User enters a title.
- (4) User enters a description.
- (5) User chooses subjects.
- (6) User clicks is public switch (optional)
- (7) User clicks on Choose file button (optional)
- (8) User clicks Add new question button

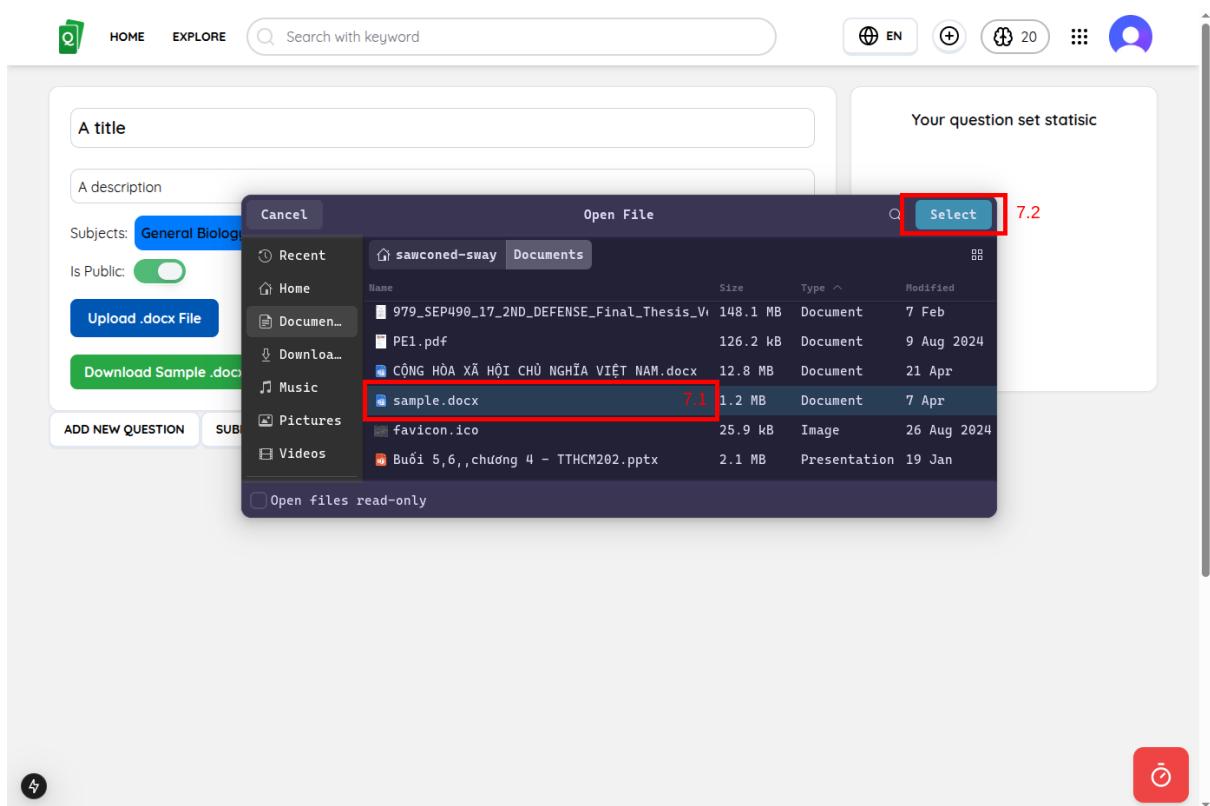


Figure 3.3.12.c: Create question sets - 3

- (7.1) User choose a file
- (7.2) User clicks Select
- After select a file, the content will be automatically added as new questions

The screenshot shows the Quizzzee application interface for creating a question set. At the top left, there are buttons for 'Is Public' (switched on), 'Upload .docx File', and 'Download Sample .docx'. To the right is a teal circular progress bar labeled 'Questions'. Below this is a list of questions. The first question is highlighted with a red box and labeled '9'. Its type is set to 'Written' (also boxed). The second question is labeled '9.1' and is a 'Multiple choice' type with 'Written' checked. Below the questions is a toolbar with various formatting icons (H1-H3, B, I, U, etc.). A text input field for 'New question' is labeled '10' and has a red box around it. Below it is a 'new answer' input field labeled '11' with a red box. At the bottom are 'ADD NEW QUESTION' and 'SUBMIT' buttons, both with red boxes and labeled '12'.

Figure 3.3.12.d: Create question sets - 4

- (9) User clicks on question type (optional)
- (9.1) User chooses question type (optional)
- (10) User inputs Question
- (11) User inputs answer
- (12) User clicks on Submit button

3.3.13 Learner update question sets

3.3.13.1 Update question set as a whole

The screenshot shows a web-based learning platform interface. At the top, there is a navigation bar with icons for HOME, EXPLORE, and a search bar labeled "Search with keyword". On the right side of the top bar are buttons for EN, a plus sign, a gear icon with the number 20, and a user profile icon. Below the top bar, the main content area has a title "A title" and a description "About this question set: A description". It also shows "Last update: 2025-05-04", a rating of "0.0" with five green stars, and a "Tags" section containing "General Biology". To the right of this information is a vertical toolbar with icons for download, copy, and update, where the "Update" icon is highlighted with a red box. Below this toolbar, a user profile for "nta123" is shown with a "FOLLOW" button. In the center, there is a circular "Learning Progress" chart filled with orange, indicating "Not Learned". To the right of the chart are four buttons: FLASH CARD, PRACTICE, QUIZ, and PLAY. At the bottom left, a URL "localhost:3000/create-set?data=68174c94f774d6ae0689b..." is visible.

Figure 3.3.13.1.a: Update question set - 1

- (1) Inside question set detail page, user clicks the update button to be redirected to the update question set page

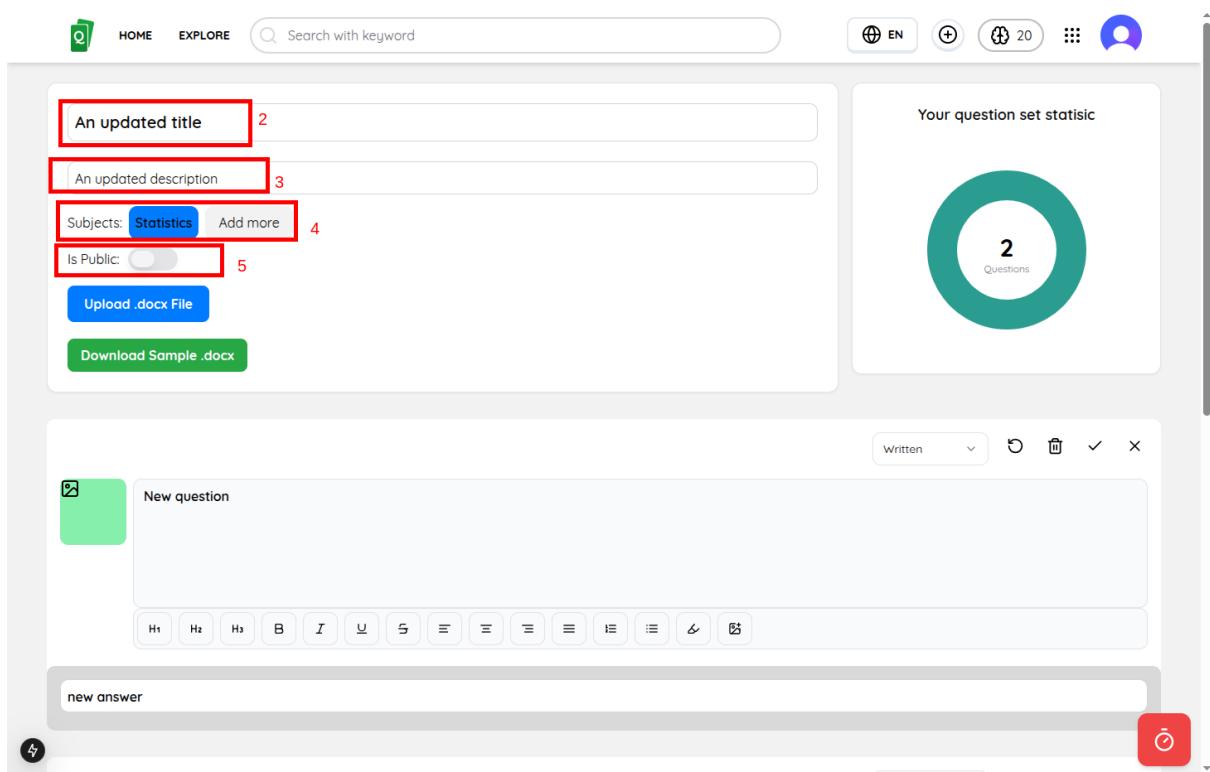


Figure 3.3.13.1.b: Update question set- 2

- (2) User updates the title
- (3) User update description
- (4) User changes subjects
- (5) User change question set visibility

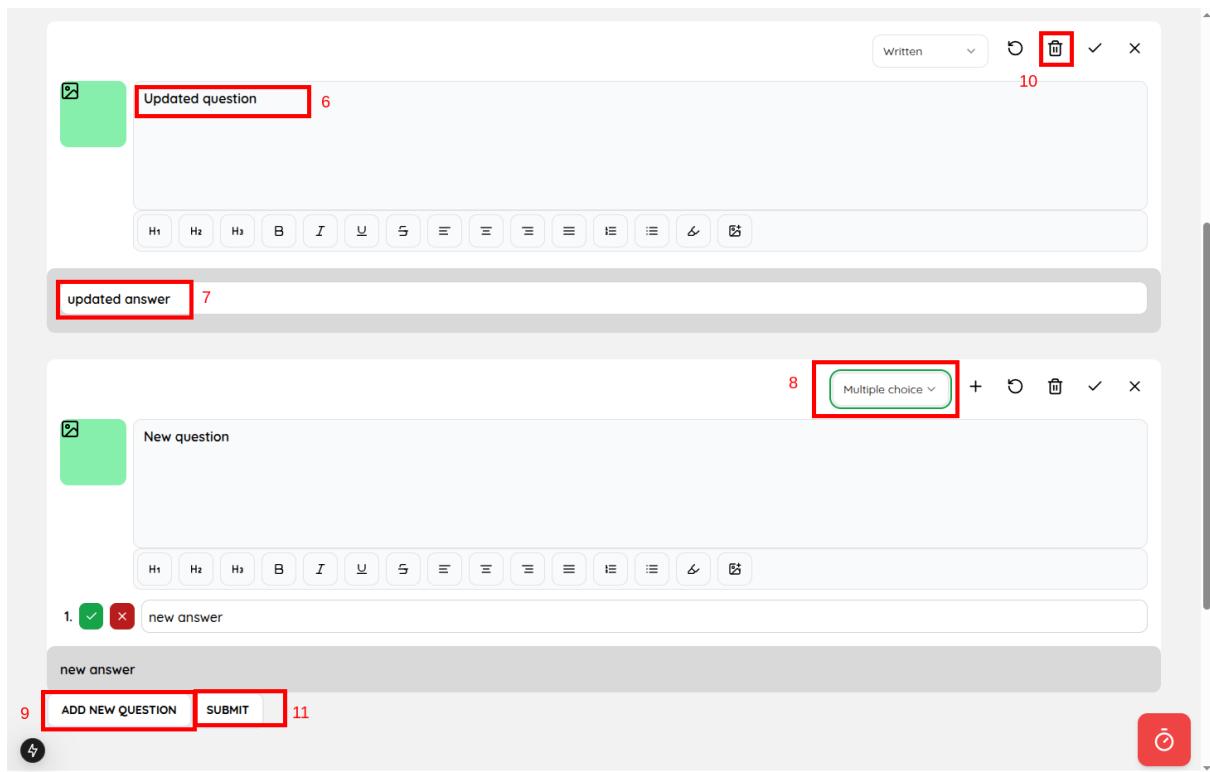


Figure 3.3.13.1.c: Update question set -3

- (6) User change question
- (7) User change answer
- (8) User change question type
- (9) User clicks Add new question to add more questions to the set
- (10) User clicks the delete icon to delete selected questions.
- (11) Once finish modifying, user clicks submit to complete update question set

3.3.13.2 Update individual question

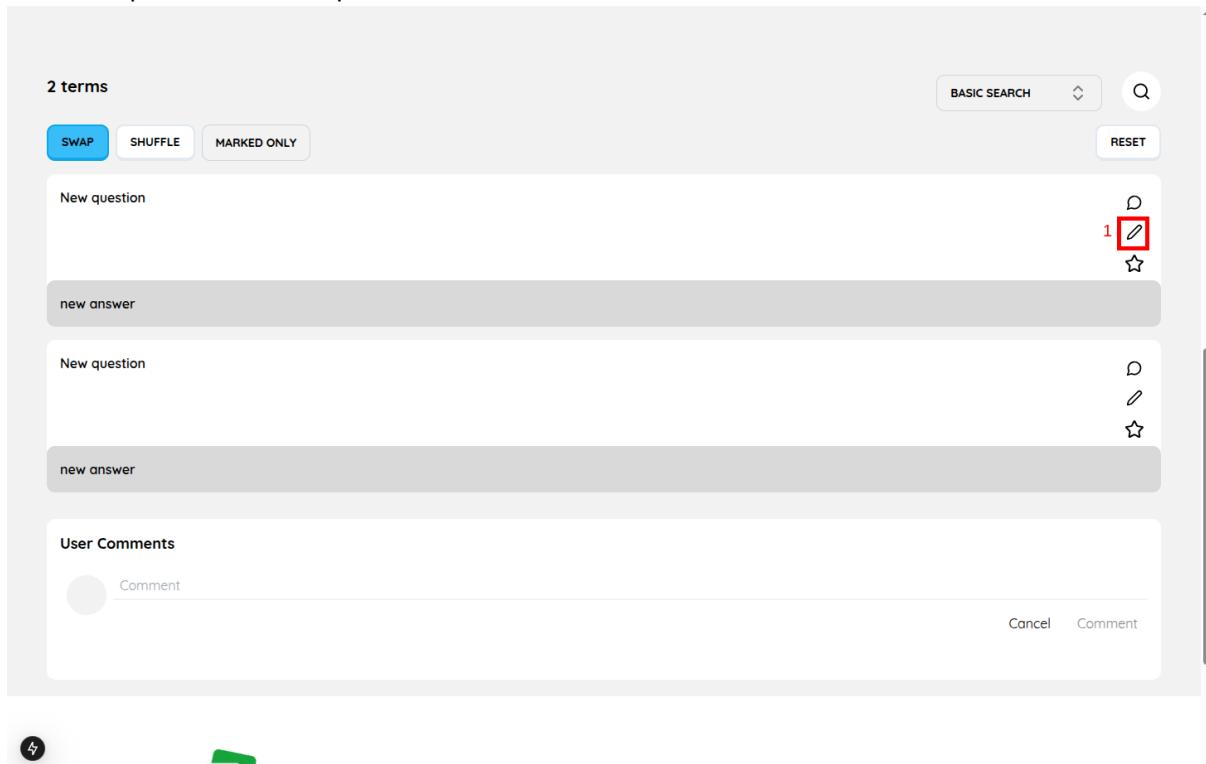


Figure 3.3.13.2.a: Update individual question - 1

- Inside the question set detail page, the user scroll to the question they want to update
- (1) User clicks on the Edit icon

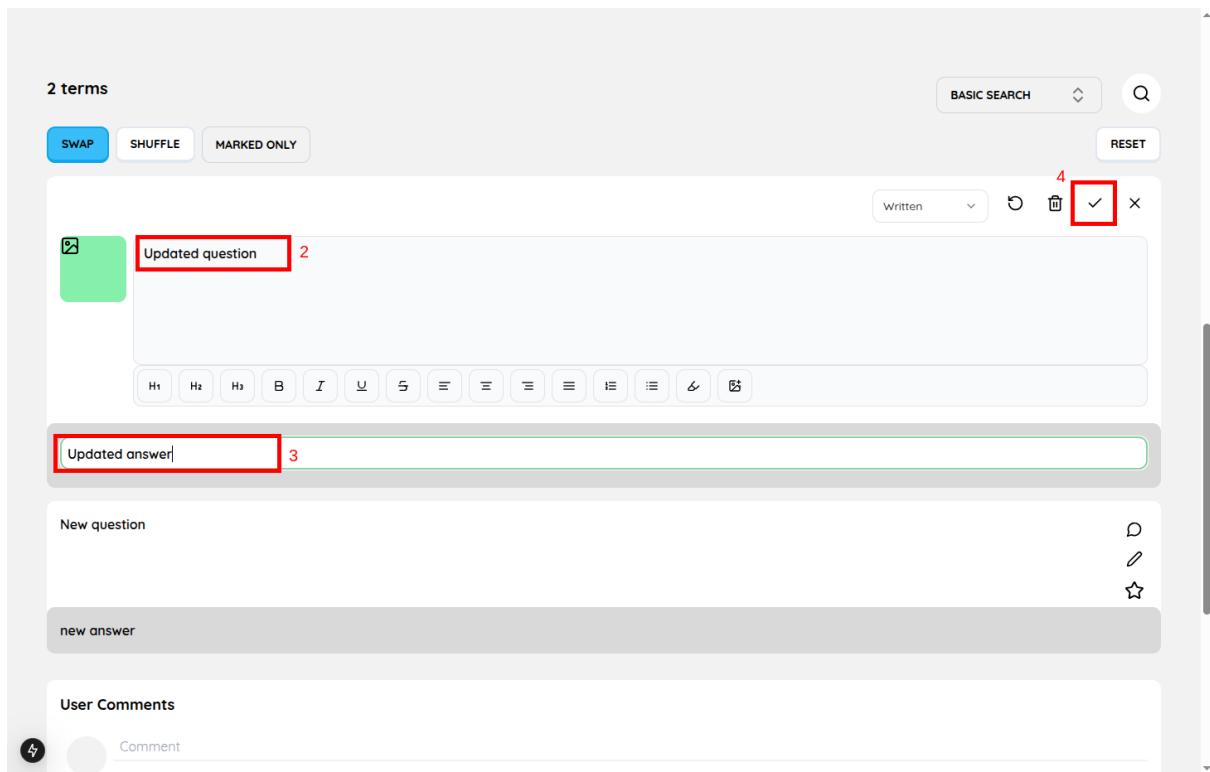


Figure 3.3.13.2.b: Update individual question - 2

- (2) User updates the question.
- (3) User updates the answer.
- (4) User clicks the check icon to complete updating a question.

3.3.14 Learner delete question sets

The screenshot shows a user interface for managing question sets. At the top, there are navigation links for 'HOME' and 'EXPLORE', a search bar, and various user account icons. Below this, the heading 'Your Question Sets' is displayed, with the sub-instruction 'Manage and organize your created and favorite question sets.' Underneath, there are two tabs: 'Created' (which is selected and highlighted with a red box labeled '1') and 'Favorite'. A search bar and a 'CREATE' button are also present. The main area lists question sets. The first set is titled 'A title' and belongs to user '@nta123' with 2 questions. It includes fields for 'A description' and 'General Biology' tags, along with a star rating. The second set is titled 'This title' and also belongs to '@nta123' with 2 questions. It includes fields for 'This description' and 'Electromagnetism' and 'General Biology' tags, along with a star rating. A modal window is overlaid on the screen, centered over the 'A title' set. It contains three buttons: 'Update', 'Delete Question card' (which is highlighted with a red box labeled '3'), and 'Cancel'. The bottom right corner of the modal has a red box around it.

Figure 3.3.14: Delete question set

- (1) Inside the sets page, make sure the user selects the created tab.
- (2) User clicks on the more icon.
- (3) User clicks on the delete button.
- (4) User clicks on the confirm modal to complete deleting the question set.

3.3.15 Share question sets

3.3.15.1 Learner and Guest copy Question set URL

The screenshot shows a user interface for a question set. At the top, there is a navigation bar with icons for HOME, EXPLORE, and a search bar labeled "Search with keyword". On the right side of the top bar are buttons for EN, +, a notification icon (20), and a user profile icon. Below the top bar, the main content area has a title "A title" and a description "About this question set: A description". It also shows the last update date (2025-05-04), a rating of 0.0 with four green stars, and a tag "Tags: General Biology". To the right of this information is a vertical menu with icons for download, share (highlighted with a red box and the number 1), embed, and favorite. Below this menu, there is a user profile section for "nta123" with a "FOLLOW" button. Further down, there is a "Learning Progress" chart with a large orange circle, a legend indicating "Not Learned" (orange), "Learning" (green), "Almost done" (dark blue), and "Mastered" (yellow), and four buttons: FLASH CARD, PRACTICE, QUIZ, and PLAY.

Figure 3.3.15.1.a: Share question set - 1

- (1) Inside the question set detail page, the user clicks the “Share” button.

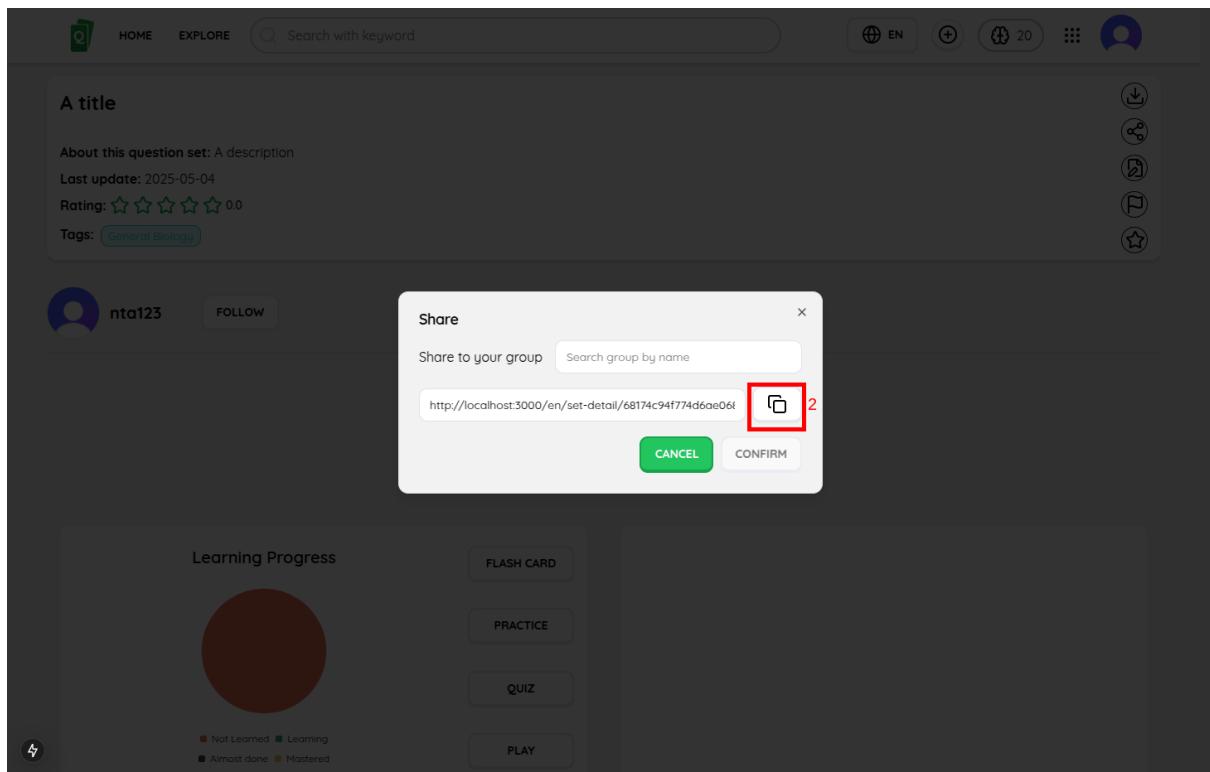


Figure 3.3.15.1.b: Share question set - 2

- (2) User clicks the copy icon to copy the URL to the system clipboard.

3.3.15.2 Learner share question set to group

The screenshot shows a user interface for a question set. At the top, there are navigation links for 'HOME' and 'EXPLORE', a search bar, and various user account icons. On the right side of the header, there is a red box highlighting the 'Share' button, which is located next to a download icon. Below the header, the main content area displays a title 'A title', a description 'About this question set: A description', a last update date '2025-05-04', a rating of '0.0' with five green stars, and a tag 'General Biology'. Below this, a user profile for 'nta123' is shown with a 'FOLLOW' button. In the center, there is a placeholder for a user profile picture with the text 'I think this Quizzzee's rating should be...' followed by five empty star rating slots. To the left, a 'Learning Progress' section features a large orange circle representing the progress status. To the right of the progress circle are four buttons: 'FLASH CARD', 'PRACTICE', 'QUIZ', and 'PLAY'. A legend at the bottom of the progress section defines the colors: orange for 'Not Learned', teal for 'Learning', dark blue for 'Almost done', and yellow for 'Mastered'.

Figure 3.3.15.2.a: Share question set - 1

- (1) Inside the question set detail page, the user clicks the share button.

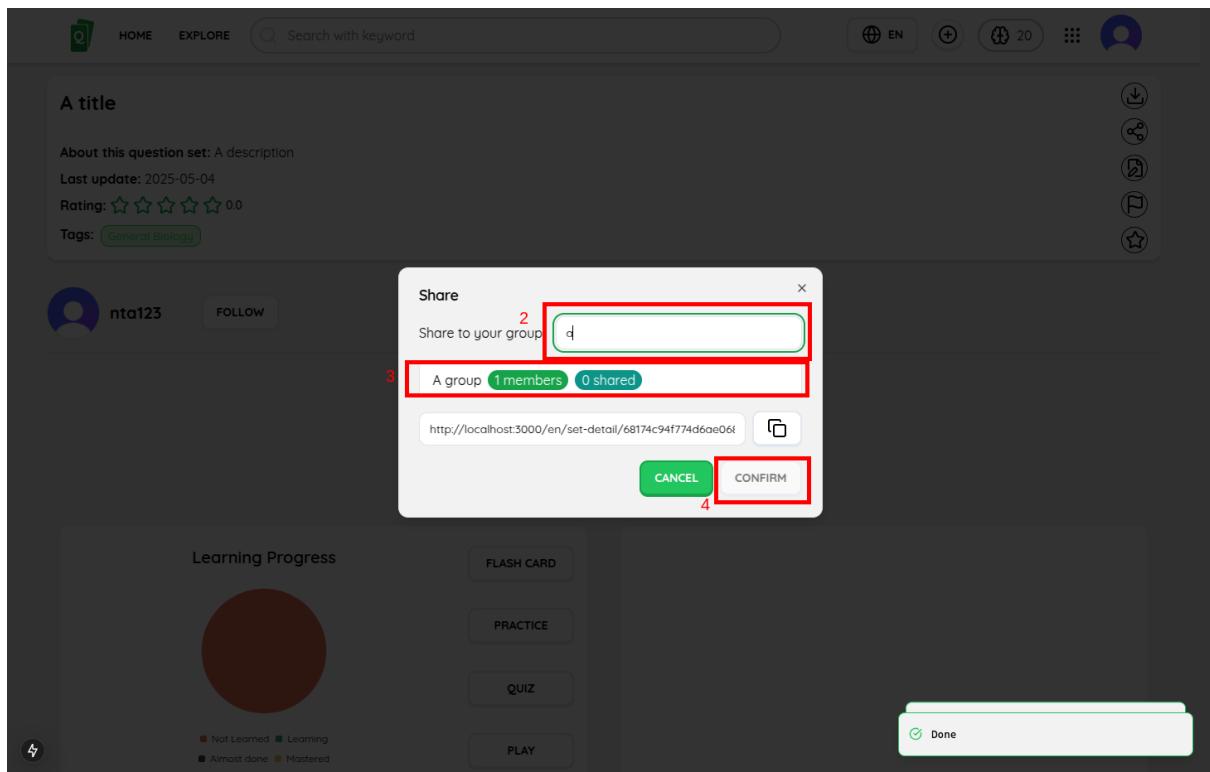


Figure 3.3.15.2.b: Share question set - 2

- (2) User input group name.
- (3) User chooses a displayed group.
- (4) User clicks the Confirm button to complete sharing the question set.

3.3.15.3 Learner share question set from group

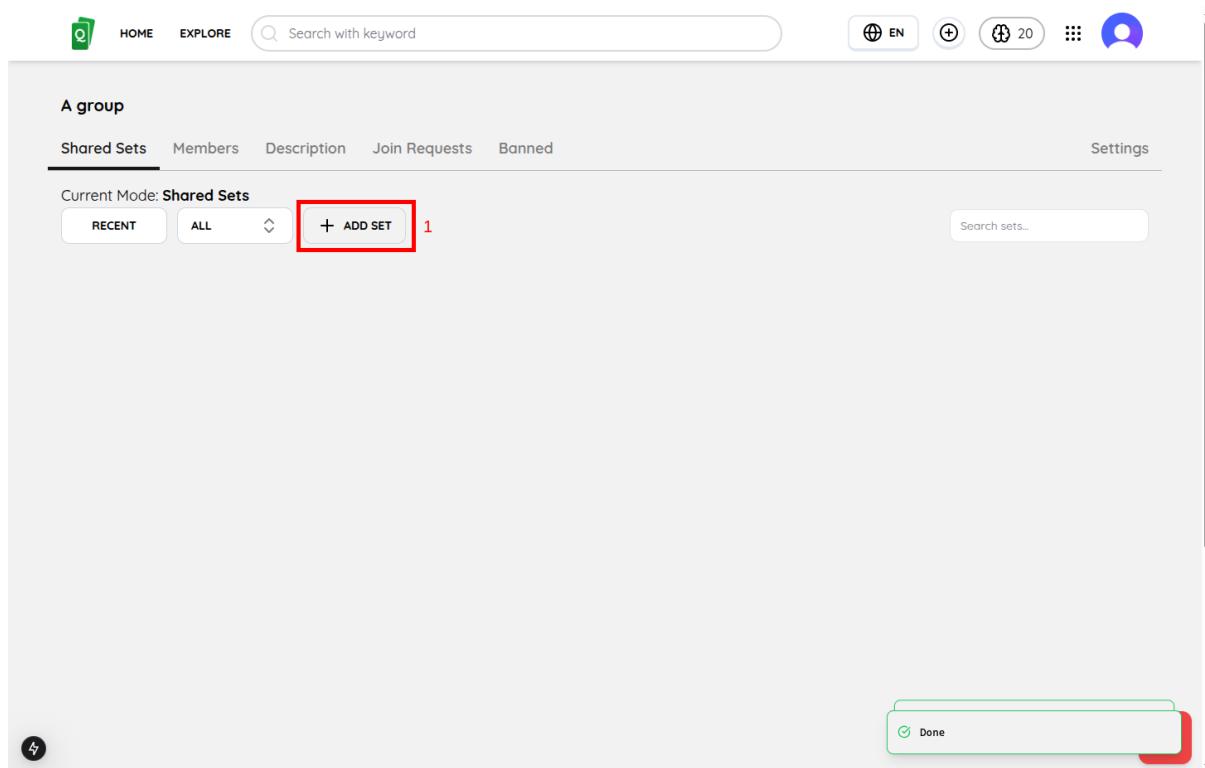


Figure 3.3.15.3.a: Share question set - 1

- (1) Inside the group detail page, the user clicks on the add set button

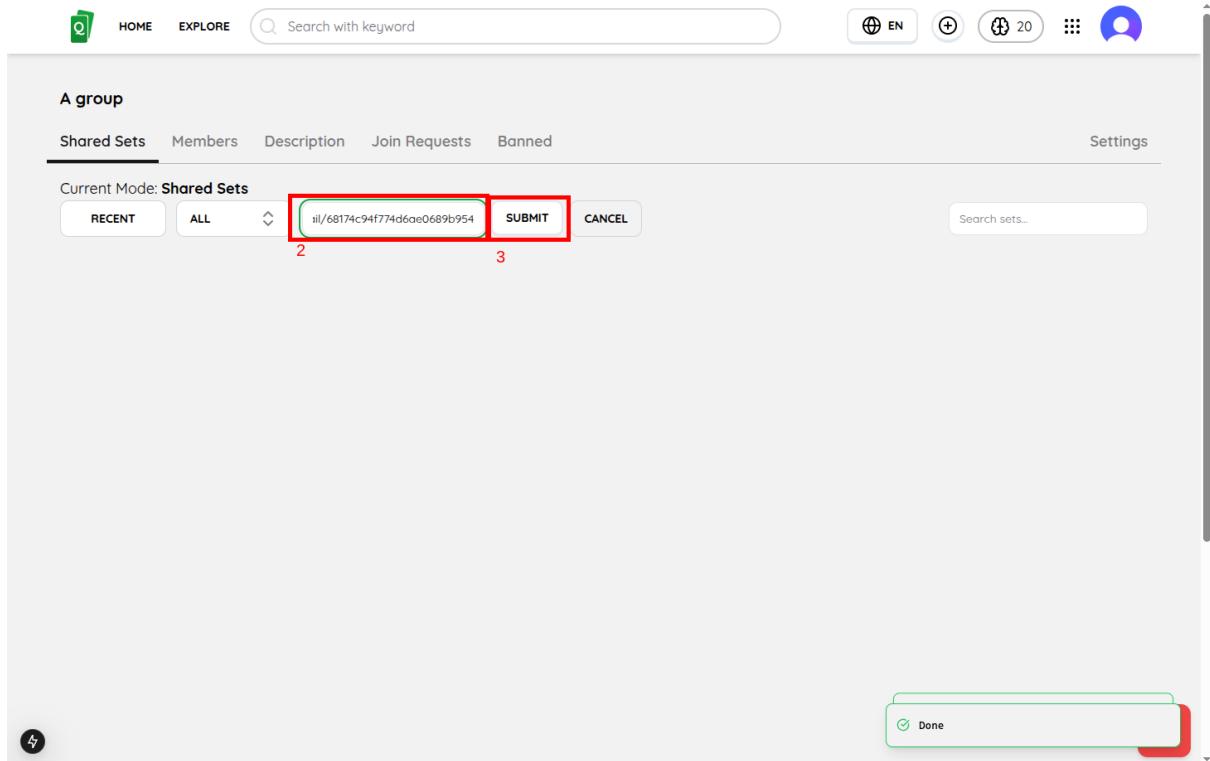


Figure 3.3.15.3.b: Share question set - 2

- (2) User inputs question set URL.
- (3) User clicks the Submit button to complete sharing the question set.

3.3.16 Learner export question set

The screenshot shows a user interface for a question set. At the top, there is a navigation bar with icons for HOME, EXPLORE, and a search bar labeled "Search with keyword". On the right side of the top bar are buttons for EN, +, 20 notifications, and a user profile icon. Below the top bar, the main content area has a title "A title" and a description "About this question set: A description". It also shows the last update date (2025-05-04), a rating of 0.0 with five green stars, and a tag "General Biology". To the right of this information is a vertical column of icons: a download button (with a red box around it), a link icon, a share icon, a refresh icon, and a star icon. Below this, there is a user profile section for "nta123" with a "FOLLOW" button. Further down, there is a "Learning Progress" chart with a large orange circle representing "Not Learned" and a small green square representing "Learning". To the right of the chart are buttons for "FLASH CARD", "PRACTICE", "QUIZ", and "PLAY". At the bottom left, a URL is visible: "localhost:3001/api/v1/questionSets/68174c94f774d6ae0689b954/export".

Figure 3.3.16.a: Export question set - 1

- (1) Inside the question set detail page, the user clicks the Export button.

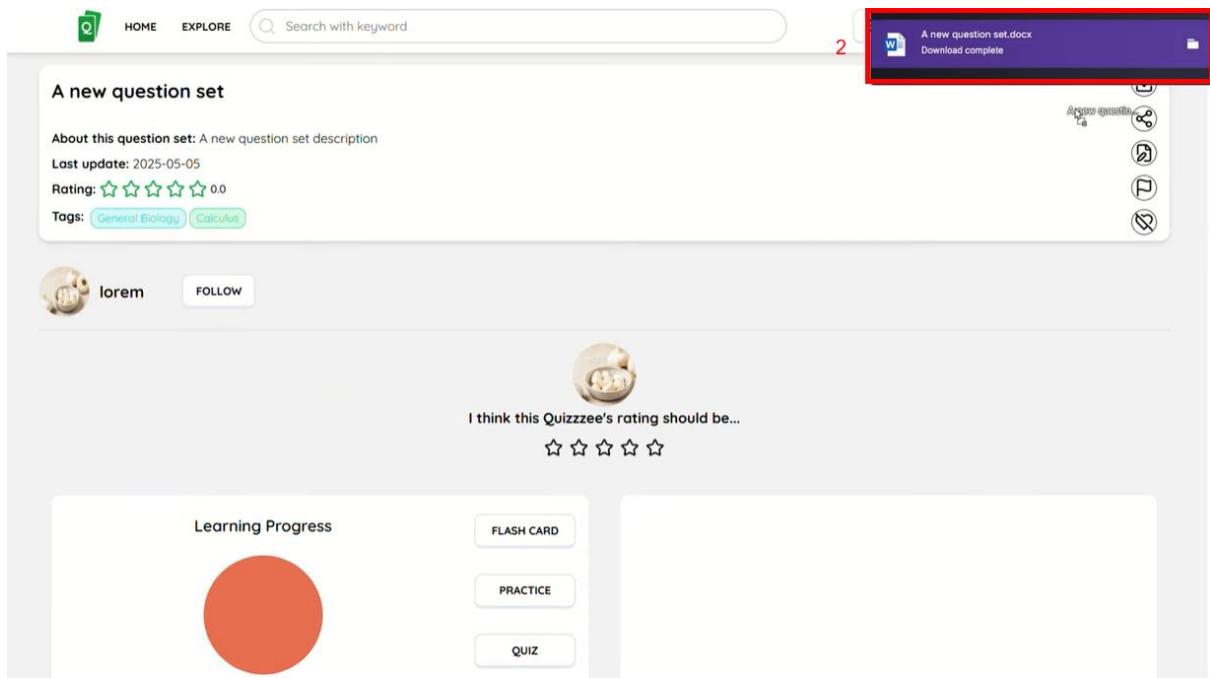


Figure 3.3.16.b: Export question set - 2

- (2) After choosing a download destination, the question set will then be downloaded into the user local machine.

3.3.17 Learner rate question set

The screenshot shows a user interface for rating a question set. At the top, there's a navigation bar with icons for HOME, EXPLORE, a search bar, and various settings like language (EN), notifications (20), and profile. Below the header, the title 'A title' is displayed. Underneath the title are sections for 'About this question set' (with a placeholder 'A description'), 'Last update' (2025-05-04), 'Rating' (0.0 with 5 green stars), and 'Tags' (General Biology). On the right side of the header, there are five circular icons with icons: a download arrow, a magnifying glass, a person, a refresh, and a star. Below this, a user profile for 'nta123' is shown with a blue circular icon and a 'FOLLOW' button. In the center, a user has rated the question set with 5 green stars, with the text 'I think this Quizzzee's rating should be 5 stars' above the stars. To the left, a 'Learning Progress' chart is shown as a red circle, with a legend below it: Not Learned (red square), Learning (green square), Almost done (dark blue square), and Mastered (yellow square). To the right of the progress chart are buttons for 'FLASH CARD', 'PRACTICE', 'QUIZ', and 'PLAY'. A green rectangular button at the bottom right is labeled 'Done' with a checkmark icon.

Figure 3.3.17: Rate question set

- Inside the question set detail page, the user clicks on one of the star icons to complete the rating.

3.3.18 Learner toggle favourite

The screenshot shows a user interface for a question set detail page. At the top, there is a navigation bar with icons for HOME, EXPLORE, a search bar labeled 'Search with keyword', and various user account and settings icons. Below the navigation bar, the title of the question set is 'A title'. Underneath the title, there is descriptive text: 'About this question set: A description', 'Last update: 2025-05-04', 'Rating: ★★★★ 5.0', and 'Tags: General Biology'. To the right of this information is a red-bordered 'Favorite' button with a heart icon. Further down the page, there is a section titled 'Learning Progress' featuring a large orange circle representing the progress status. To the right of the progress circle are four buttons: 'FLASH CARD', 'PRACTICE', 'QUIZ', and 'PLAY'. On the far right, there is a green-bordered 'Done' button with a checkmark icon. At the bottom of the page, there is a user profile for 'nta123' with a 'FOLLOW' button, and a rating section asking 'I think this Quizzzee's rating should be...' with five stars.

Figure 3.3.18: Toggle favourite

- Inside the question set detail page, the user clicks on the heart icon to set a question set as favorite.
- Once a set is added to favourite, clicking the heart icon again will un-favourite question set.

3.3.19 Learner mark questions

The screenshot shows a user interface for managing questions. At the top, there are buttons for 'SWAP', 'SHUFFLE', and 'MARKED ONLY'. A search bar labeled 'BASIC SEARCH' with a magnifying glass icon and a 'RESET' button are also present. Below these, there are two sections, each containing a 'New question' and a 'new answer' field. The first section has a yellow star icon at the top right, which is highlighted with a red box. The second section has a grey star icon. At the bottom, there is a 'User Comments' section with a placeholder 'Comment' and 'Cancel' and 'Comment' buttons.

Figure 3.3.19: Mark questions

- Inside the question set detail page, the user clicks the star icon inside a question to complete marking the question.

3.3.20 Learner and Guest Filter questions

3.3.20.1 Learner filter marked questions

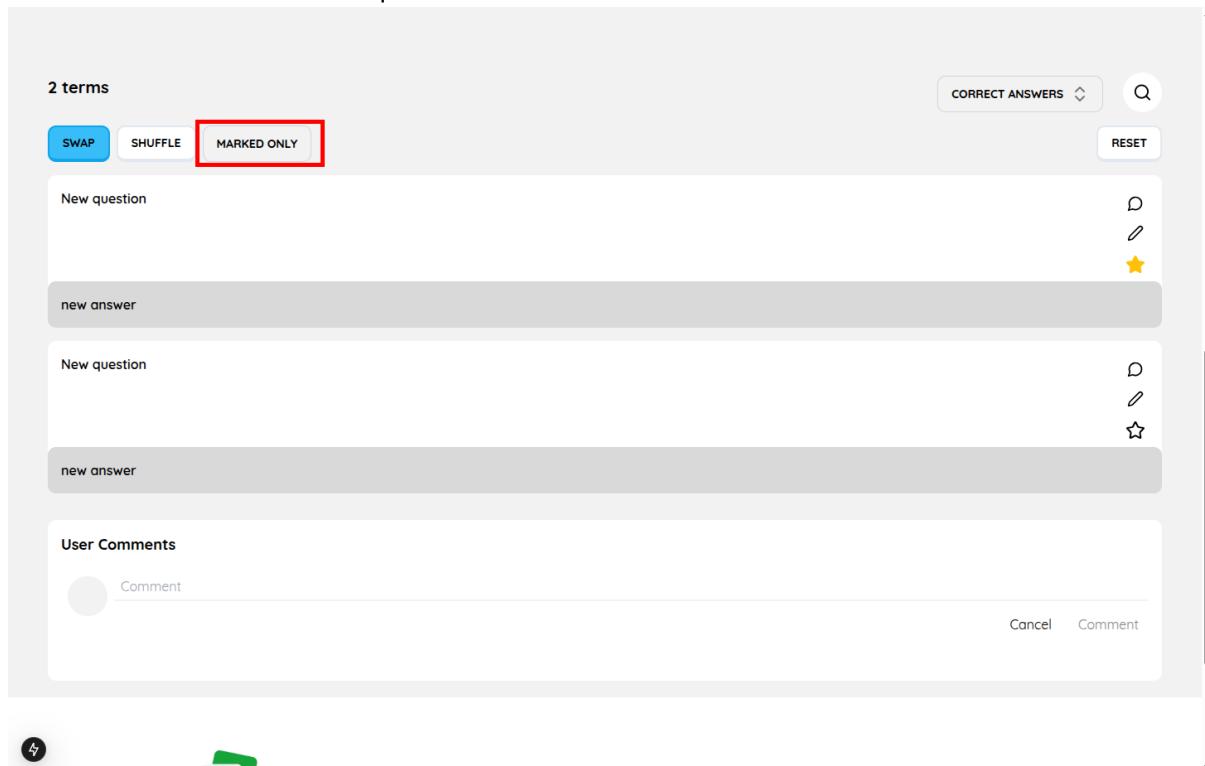


Figure 3.3.20.1: Filter marked questions

- Inside the question set detail page, the user clicks on marked button to complete filtering questions,

3.3.20.2 Filter with keyword

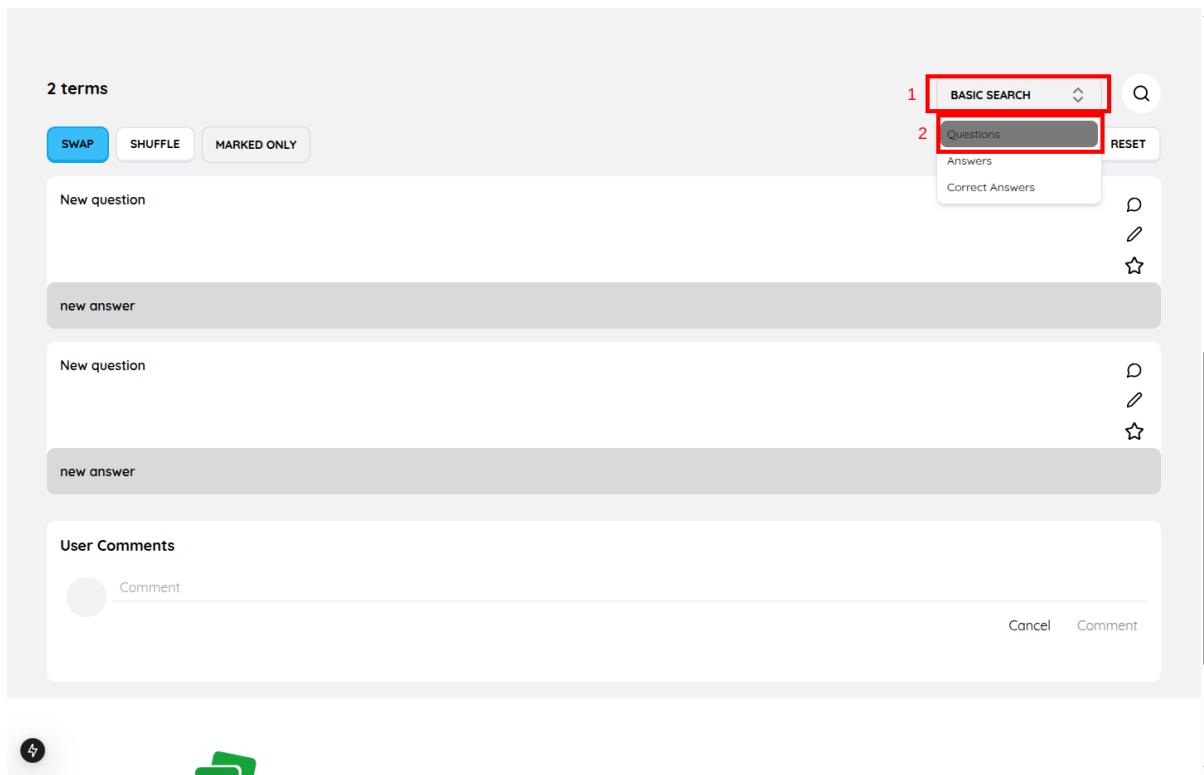


Figure 3.3.20.2.a: Filter questions - 1

- (1) Inside the question set detail page, the user clicks on the search dropdown button (optional)
- (2) User clicks on one of the options.

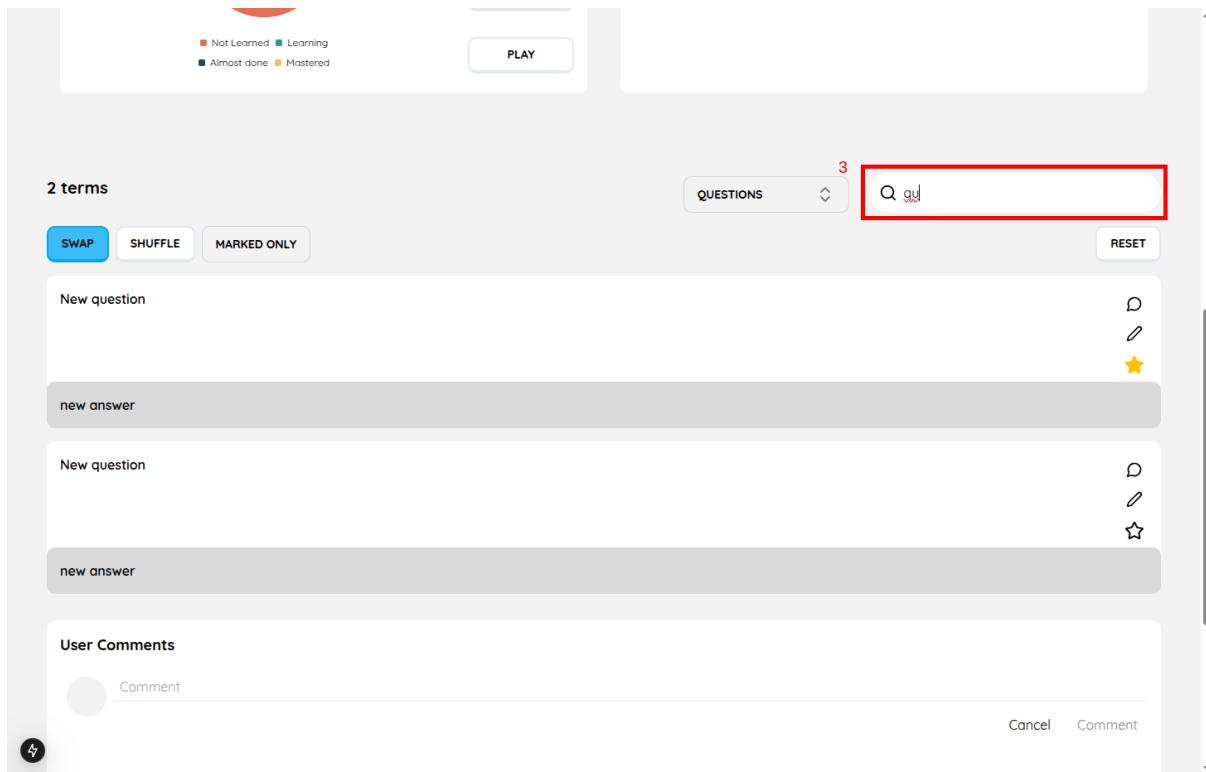


Figure 3.3.20.2.b: Filter questions - 2

- (3) User input keyword to filter for relevant questions.

3.3.21 View comments

3.3.21.1 Learner and Guest view question set comments

The screenshot shows a mobile application interface for a question set. At the top, there is a navigation bar with a back arrow, a search icon, and a star icon. Below the navigation bar is a grey header bar with the text "new answer". The main content area is titled "User Comments". It displays two comments from a user named "nta123": one from May 4, 2025, stating "I had an exam tomorrow..." with 0 replies, and another from May 4, 2025, stating "this is good!" with 0 replies. There are "Cancel" and "Comment" buttons at the bottom right of the comment section. At the bottom of the screen, there is a footer with the Quizzzee logo, the tagline "Helping you master knowledge, one card at a time", and links for "Home", "Explore", and "Profile". On the right side of the footer, there is a "Contact us" section with icons for a phone (number 0123456789) and an envelope (email quizzzee192@gmail.com). The footer also includes a copyright notice: "© 2025 Quizzzee. All rights reserved." and a small circular icon.

Figure 3.3.21.1: View comments

- Inside the question set detail page, the user scrolls down to the comment section to view comments.

3.3.21.2 Learner and Guest view question comments

The screenshot shows a user interface for managing terms. At the top, there are buttons for 'SWAP', 'SHUFFLE', 'MARKED ONLY', 'BASIC SEARCH' with a magnifying glass icon, and 'RESET'. Below this, there are sections for 'New question' and 'new answer'. On the right side, there are three vertical icons: a comment bubble with a red border, a pencil, and a star. The main content area is titled 'User Comments' and contains a form with a placeholder 'Comment'. Below the form, there are two comments from a user named 'nta123' dated 2025-05-04. The first comment says 'I had an exam tomorrow...' and has a 'Reply' button. The second comment says 'nice' and also has a 'Reply' button. There are also 'Cancel' and 'Comment' buttons at the bottom of the comment section.

Figure 3.3.21.2.a: View comments - 1

- Inside the question set detail page, the user clicks on the comment icon of a question to view its comments.

This screenshot shows a modal window titled 'Comment on "New question"'. The window has a dark background and displays a list of comments from 'nta123'. The first comment, 'this comment', is highlighted with a green border and labeled 'Accepted Answer'. Below it, another comment 'this comment' has a 'Reply' button. A third comment 'nice' also has a 'Reply' button. The modal includes standard UI elements like 'Comment' and 'Cancel' buttons. The background of the main application interface is visible, showing other sections like 'User Comments' and 'New question'.

Figure 3.3.21.2.b: View comments - 2

3.3.22 Learner creates comments

3.3.22.1 Create question set comment

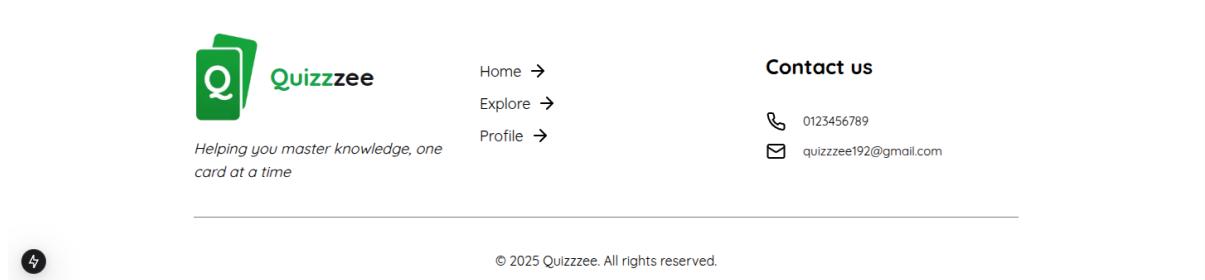
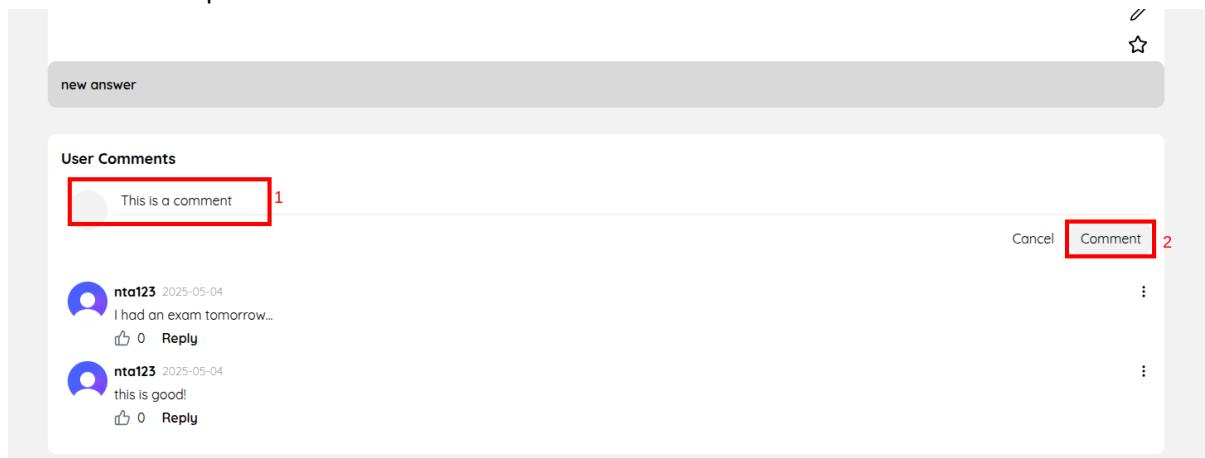


Figure 3.3.22.2: Create comments

- (1) Inside the question set detail page, the user clicks on the commenting prompt and inputs a comment.
- (2) User clicks the Comment button to complete creating a comment.

3.3.22.2 Create question comment

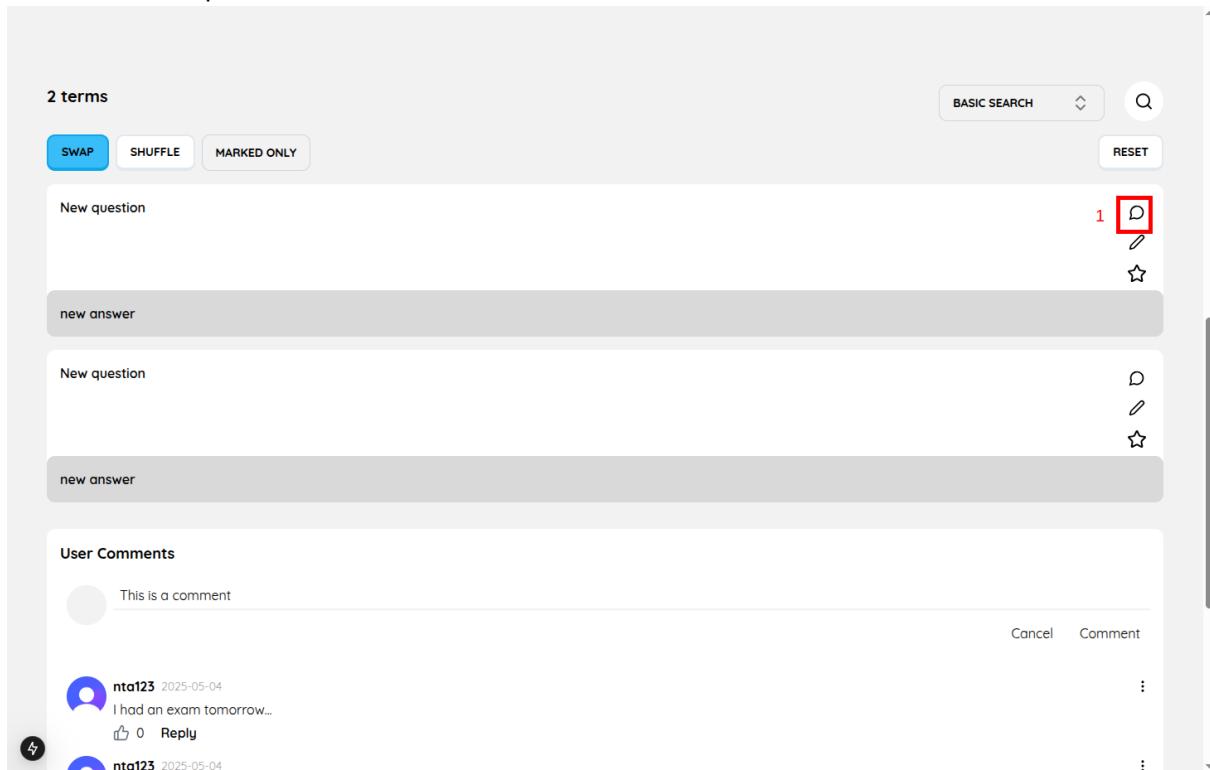


Figure 3.3.22.2.a: Create comments - 1

- (1) Inside the question set detail page, the user clicks on the comment button inside the question card.

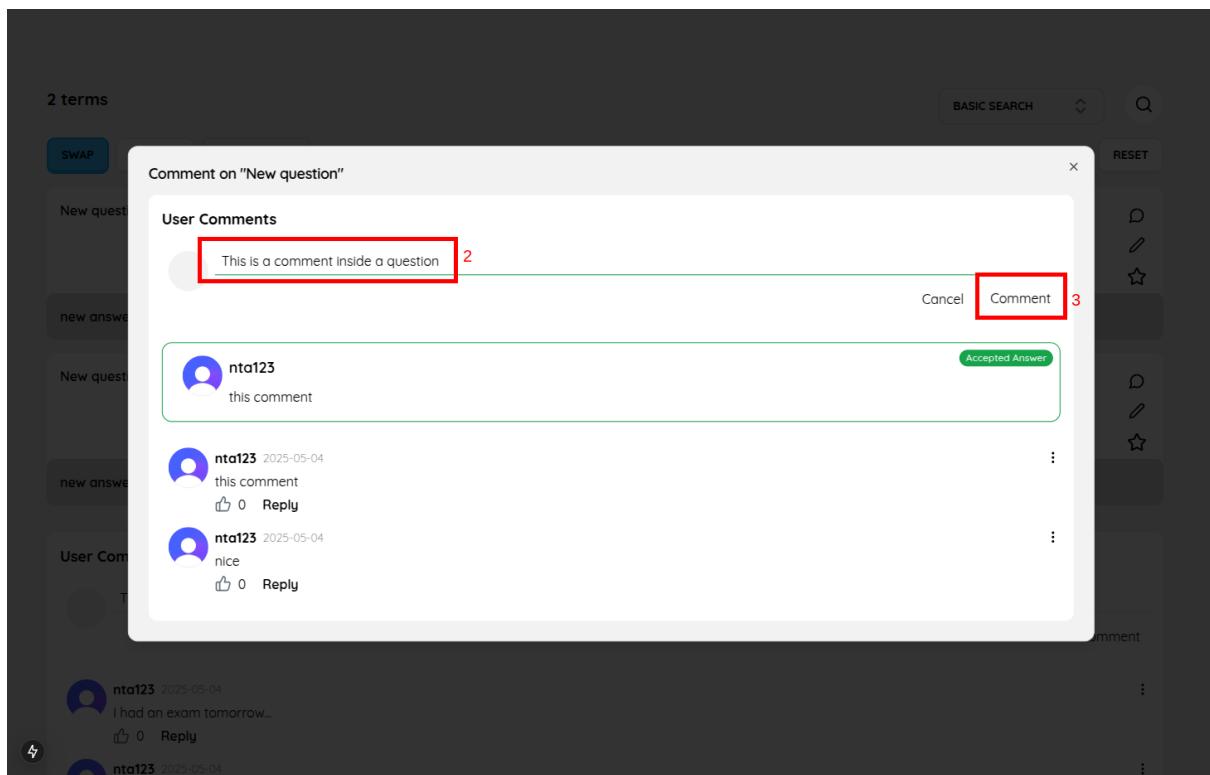


Figure 3.3.22.2.b: Create comments - 2

- (2) User clicks on the commenting prompt and inputs a comment.
- (3) User clicks the Comment button to complete creating a comment.

3.3.23 Learner reply comments

3.3.23.1 Reply question set comments

The screenshot shows a mobile application interface for 'User Comments'. At the top, there is a 'new answer' button and a star icon. Below this, a 'User Comments' section displays two comments from a user named 'nta123' dated 2025-05-04. The first comment says 'I had an exam tomorrow...' with 0 upvotes and a 'Reply' button (labeled 1). A reply card is open below it, containing the text 'Good luck dude!' with 2 upvotes and a 'Reply' button (labeled 3). The second comment simply says 'this is good!' with 0 upvotes and a 'Reply' button. At the bottom right of the screen, there is a 'Contact us' section with icons for phone and email, and contact information: 0123456789 and quizzzee192@gmail.com.

new answer

User Comments

Comment

Cancel Comment

nta123 2025-05-04
I had an exam tomorrow...
0 Reply 1

Good luck dude! 2

Cancel Reply 3

nta123 2025-05-04
this is good!
0 Reply

Home →
Explore →
Profile →

Contact us

0123456789
quizzzee192@gmail.com

Figure 3.3.23.1: Reply comments

- (1) User clicks on the reply button inside the comment card.
- (2) User inputs a comment.
- (3) User clicks the reply button to complete replying to a comment.

3.3.23.2 Reply question comments

The screenshot shows a digital application interface with the following elements:

- Top Bar:** Includes a "BASIC SEARCH" dropdown, a magnifying glass icon, and a "RESET" button.
- Left Side:** A vertical sidebar with buttons for "SWAP", "SHUFFLE", and "MARKED ONLY".
- Content Area:** A list of items:
 - New question**: Has a red box around its comment icon (a speech bubble with a '1').
 - new answer**: Has a star icon.
 - New question**: Has a pencil icon.
 - new answer**: Has a star icon.
- User Comments Section:** Titled "User Comments". It contains two comments from a user named "nta123":
 - nta123 2025-05-04**: "I had an exam tomorrow..." with a reply count of 0.
 - nta123 2025-05-04**: Another comment entry.
- Bottom Right:** Buttons for "Cancel" and "Comment".

Figure 3.3.23.2.a: Reply comments - 1

- (1) User clicks on the comment icon inside the question card.

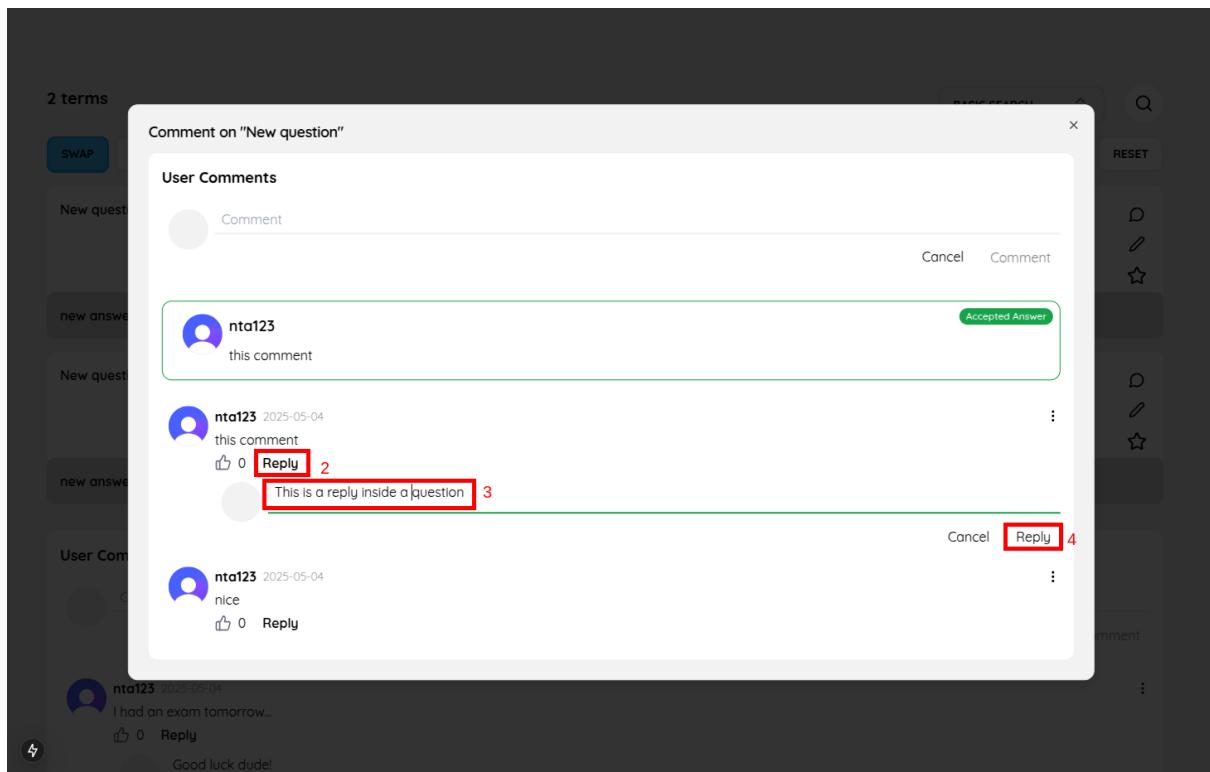


Figure 3.3.23.2.b: Reply comments - 2

- (2) User clicks on the reply button inside the comment card.
- (3) User inputs a comment.
- (4) User clicks the reply button to complete replying to a comment.

3.3.24 Learner update comments

3.3.24.1 Update question set comments

The screenshot shows a mobile application interface for a question set. At the top, there is a list of terms: "3. Neutron" and "4. Photon". Below this, the word "Proton" is highlighted in a grey bar. A button labeled "SHOW 5 MORE TERMS" is visible. The main content area is titled "User Comments" and shows two comments from a user named "nta123" dated 2025-05-09. Each comment states: "This question might be too easy for the difficulty level" and has a "Reply" button. To the right of the comments, there is a "Newest comments" section with a "Cancel" and "Comment" button. A modal window is open over the comments, containing a list of options: "1 ⋮", "2 Edit Comment", and "Delete Comment". The "Edit Comment" option is highlighted with a red box. At the bottom of the screen, there is a navigation bar with icons for Home, Explore, Profile, and Contact us. The Quizzzee logo is also present.

Figure 3.3.24.1.a: Update comments - 1

- (1) User clicks on the more button.
- (2) User clicks on the edit icon inside the comment card.

The screenshot shows a user interface for updating a comment. At the top, there is a list of terms: "3. Neutron" and "4. Photon". Below this, a search bar contains the text "Proton". A button labeled "SHOW 5 MORE TERMS" is visible. The main area is titled "User Comments" and shows a comment from a user named "nto123" dated "2025-05-09". The comment text is "Updated comment" (highlighted with a red box). To the right of the comment are "Cancel" and "Comment" buttons. Below the comment, there is a reply from the same user: "This question might be too easy for the difficulty level" with a "Reply" button. At the bottom right of the comment card is a "SAVE" button (also highlighted with a red box). A red number "3" is placed next to the updated comment text, and a red number "4" is placed next to the "SAVE" button.

● Figure 3.3.24.1.b: Update comments - 2

- (3) User updates comment.
- (4) User clicks on the update button to complete updating a comment.

3.3.24.2 Update question comments

The screenshot shows a digital application interface with the following elements:

- Top Bar:** Includes "BASIC SEARCH" with a dropdown arrow and a magnifying glass icon, and a "RESET" button.
- Card Header:** Shows "2 terms". Below it are three buttons: "SWAP" (highlighted in blue), "SHUFFLE", and "MARKED ONLY".
- Card 1:** Labeled "New question". It has a red box around the "Comment" icon (a speech bubble with a '1'). Other icons include a pencil and a star.
- Card 2:** Labeled "new answer". It has a pencil and a star icon.
- Card 3:** Labeled "New question". It has a pencil and a star icon.
- Card 4:** Labeled "new answer". It has a pencil and a star icon.
- User Comments Section:** Labeled "User Comments".
 - A placeholder card says "This is a comment".
 - A comment by user "nta123" (2025-05-04) reads "I had an exam tomorrow...". It has a "Reply" button with "0" replies and a more options menu.
 - A second comment by user "nta123" (2025-05-04) is partially visible.
- Bottom Buttons:** "Cancel" and "Comment".

Figure 3.3.24.2.a Update question comments - 1

- (1) User clicks on the comment button inside the question card

The screenshot shows a mobile application interface for managing terms. At the top, there are buttons for 'SWAP', 'SHUFFLE', 'MARKED ONLY', 'BASIC SEARCH', and 'RESET'. Below this, a question is displayed: "Which of the following is the formula for kinetic energy?". Four options are listed: 1. $\frac{1}{2}mv^2$, 2. mgh , 3. $F=ma$, and 4. $P=IV$. The first option is highlighted with a blue background. To the right of the question are icons for a comment, a star, and a share symbol. A modal window titled "Comment on 'Which of the following is the formula for kinetic energy?'" is open, showing "User Comments" and "Newest comments". It contains a comment from user "nta123" dated 2025-05-09: "This question might be too easy for the difficulty level". Below the comment are "Cancel", "Comment", and three buttons numbered 1, 2, and 3. Buttons 2 and 3 are highlighted with red boxes. Button 2 contains a vertical ellipsis (...). Button 3 contains the text "Edit Comment". A "Delete Comment" button is also visible. The background of the app shows other questions and their options.

Figure 3.3.24.2.b Update question comments - 2

- (2) User clicks on the more button
- (3) User clicks on the update button

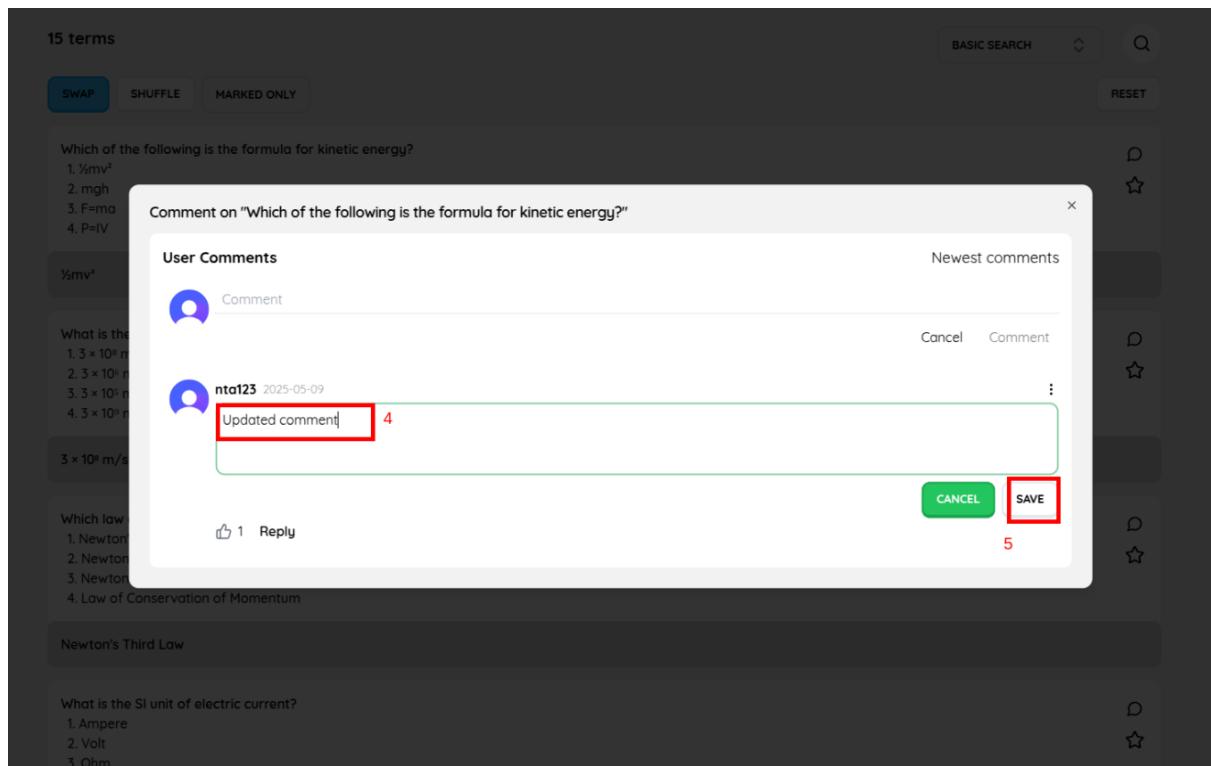


Figure 3.3.24.2.c Update question comments - 3

- (4) User updates the comment
- (5) User clicks on the update button to complete updating a comment

3.3.25 Learner delete comments

3.3.25.1 Delete question set comments

The screenshot shows a user interface for managing comments on a question set titled "Proton". At the top, there are two options: "3. Neutron" and "4. Photon". Below this is a "User Comments" section. It displays two comments from a user named "nta123" dated 2025-05-09. Each comment includes a "Reply" button. To the right of the comments is a "Newest comments" section with "Cancel" and "Comment" buttons. A dropdown menu is open over the second comment, with the "Delete Comment" option highlighted and surrounded by a red box. The Quizzzee logo and tagline "Helping you master knowledge, one card at a time" are visible at the bottom left. On the right, there are links for "Home", "Explore", and "Profile", along with contact information: a phone icon with the number 0123456789 and an email icon with the address quizzzee192@gmail.com.

Figure 3.3.25.1.a: Delete comments - 1

- (1) User clicks on the dropdown menu inside the comment card.
- (2) User clicks the Delete comment option

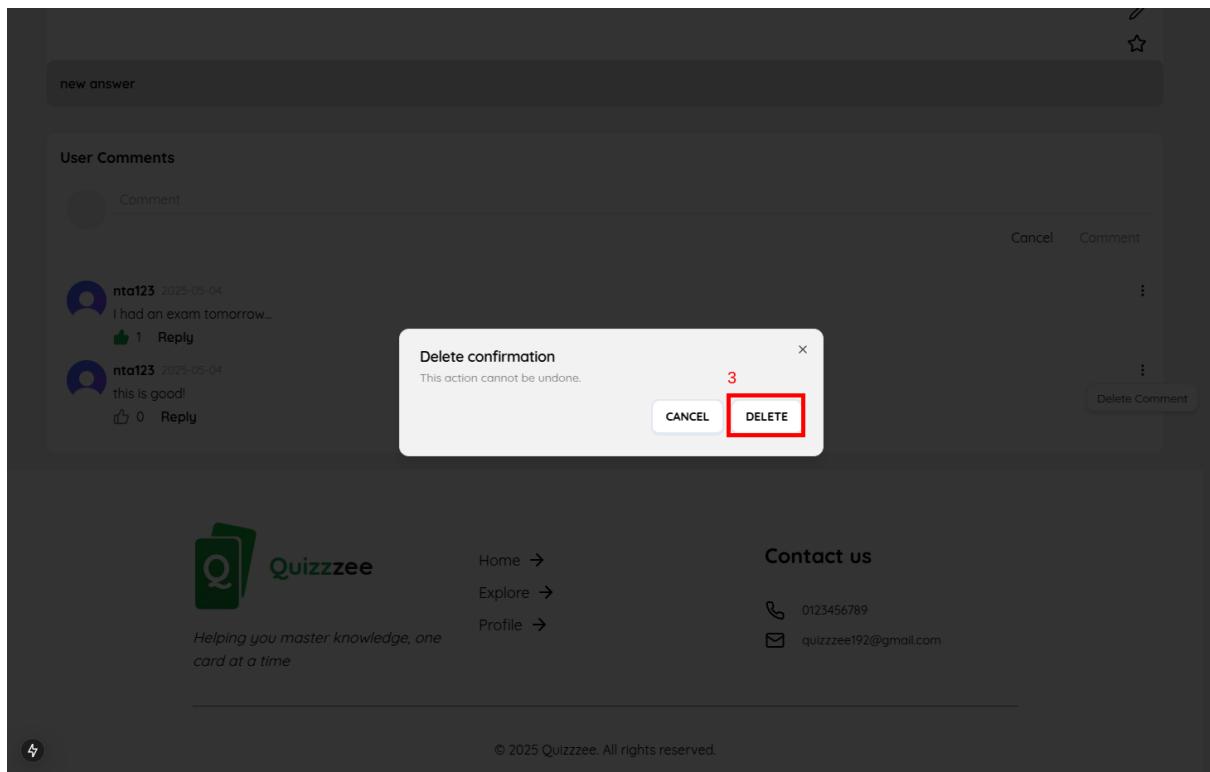


Figure 3.3.25.1.b: Delete comments - 2

- (3) User clicks on the Delete button to complete deleting a comment.

3.3.25.2 Delete question comments

The screenshot shows a digital application interface with the following elements:

- Top Bar:** Includes a "BASIC SEARCH" dropdown, a magnifying glass icon, and a "RESET" button.
- Left Side Buttons:** "SWAP", "SHUFFLE", and "MARKED ONLY".
- Item List:** Contains two items labeled "New question" and "new answer". Each item has a small red number "1" followed by a comment icon (a speech bubble with a person icon) which is highlighted with a red box.
- User Comments Section:** Titled "User Comments". It displays two comments from a user named "nta123":
 - The first comment was posted on "2025-05-04" and says "I had an exam tomorrow...". It includes a reply icon (a speech bubble with a person icon) and a "Reply" button.
 - The second comment was also posted on "2025-05-04" and is partially visible.
- Bottom Right Buttons:** "Cancel" and "Comment".

Figure 3.3.25.2.a: Delete comments - 1

- (1) User clicks on the comment icon inside the question card.

The screenshot shows a mobile application interface for a quiz or study session. At the top, there are buttons for 'SWAP', 'SHUFFLE', 'MARKED ONLY', 'BASIC SEARCH', and 'RESET'. Below this, a question is displayed: "Which of the following is the formula for kinetic energy?". Four options are listed: 1. $\frac{1}{2}mv^2$, 2. mgh , 3. $F=ma$, and 4. $P=IV$. The correct answer is $\frac{1}{2}mv^2$. A user comment is shown: "This question might be too easy for the difficulty level" by user 'nta123' on 2025-05-09. Below the comment, there are 'Cancel' and 'Comment' buttons. A dropdown menu is open with three options: 'Edit Comment' (option 2), 'Delete Comment' (option 3), and a third option whose icon is obscured by a red box. The 'Delete Comment' option is also highlighted with a red box. The background shows other questions and answers, such as "What is the SI unit of electric current?" with options 1. Ampere, 2. Volt, and 3. Ohm.

Figure 3.3.25.2.b: Delete comments - 2

- (2) User clicks on the dropdown menu inside the comment card.
- (3) User clicks the Delete comment option

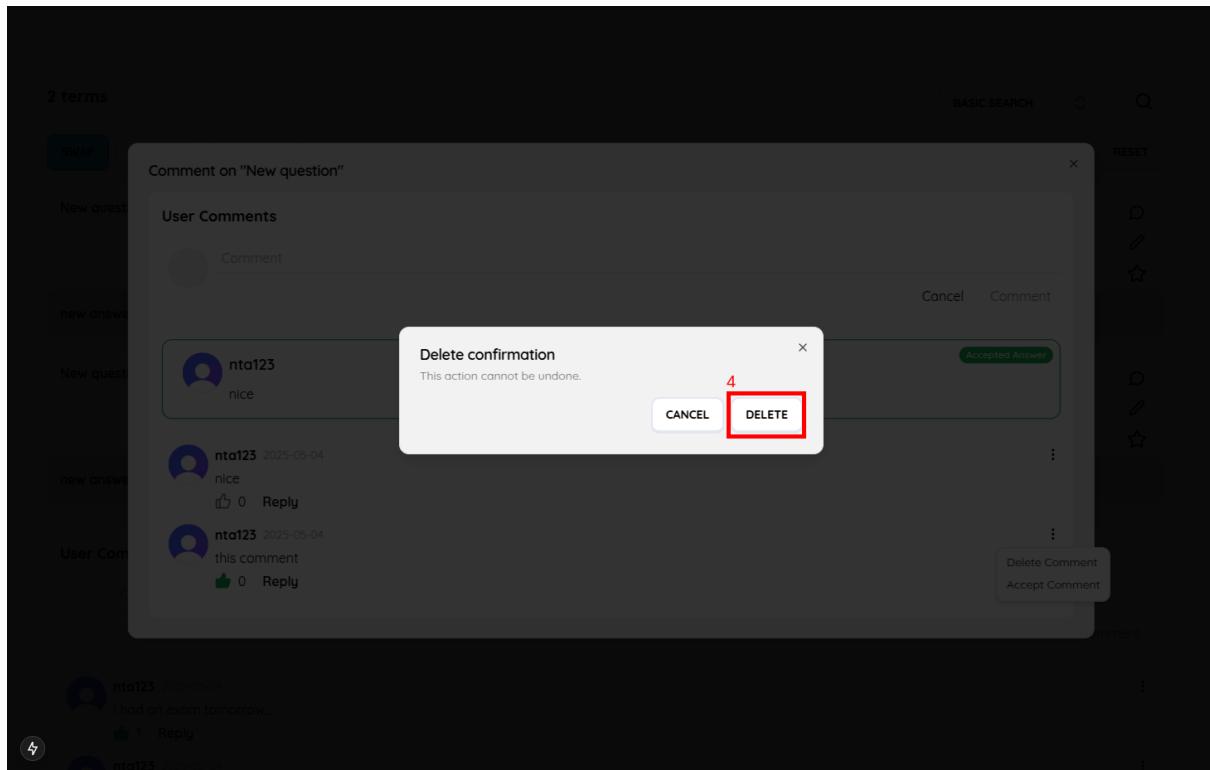


Figure 3.3.25.2.c: Delete comments - 3

- (4) User clicks on the Delete button to complete deleting a comment.

3.3.26 Learner upvote comments

3.3.26.1 Upvote question set comments

The screenshot shows a mobile application interface for 'User Comments'. At the top, there is a grey bar with a 'new answer' button and three icons: a pencil, a star, and a double arrow. Below this is a white header with 'User Comments' and a 'Comment' button. On the right side of the header are 'Cancel' and 'Comment' buttons. The main content area displays two comments from a user named 'nta123' dated 2025-05-04. The first comment says 'I had an exam tomorrow...' with a green upvote icon showing '1' and a 'Reply' button. The second comment says 'this is good!' with a green upvote icon showing '0' and a 'Reply' button. The bottom of the screen features the Quizzzee logo (a green 'Q' icon) and the text 'Helping you master knowledge, one card at a time'. To the right, there are links for 'Home →', 'Explore →', and 'Profile →'. On the far right, there is a 'Contact us' section with a phone icon and number '0123456789', and an email icon and address 'quizzzee192@gmail.com'. A copyright notice '© 2025 Quizzzee. All rights reserved.' is at the very bottom.

Figure 3.3.26.1: Upvote comments

- (1) User clicks on the upvote icon inside the comment card to complete upvoting a comment.
- If a comment is already voted, clicking the icon again will unvote the comment

3.3.26.2 Upvote question comments

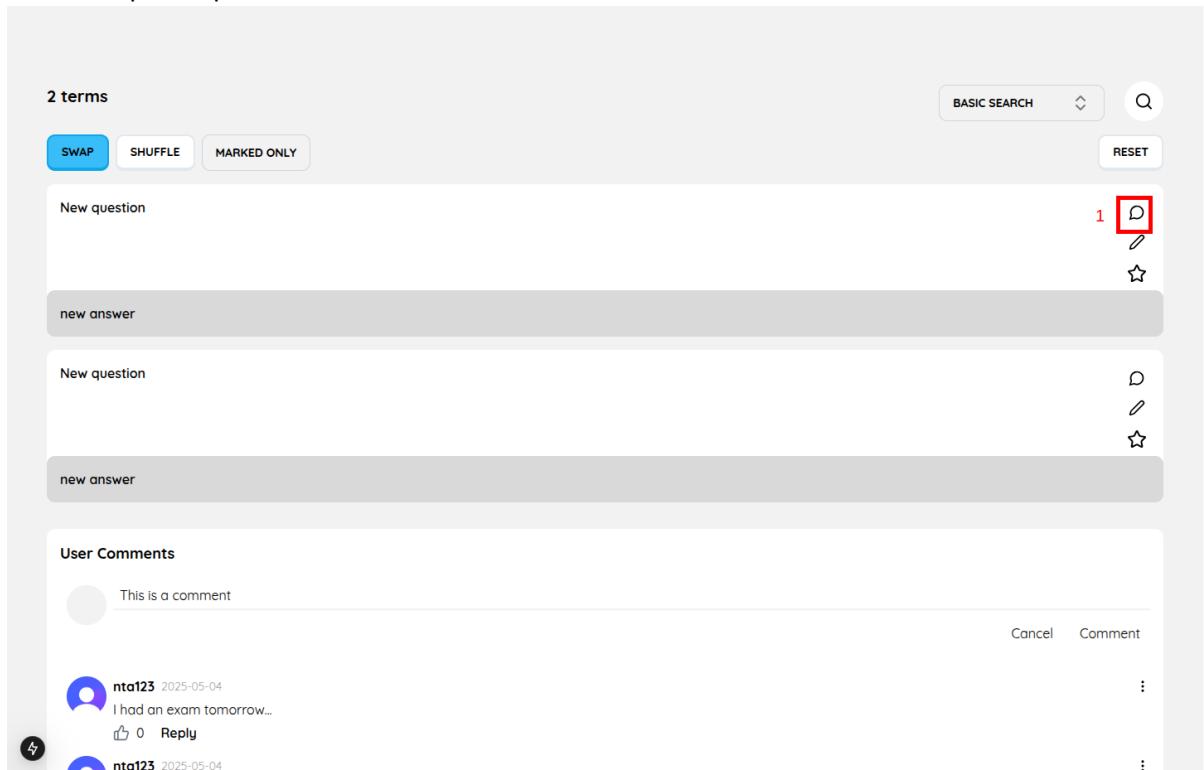


Figure 3.3.26.2.a: Upvote comments - 1

- (1) User clicks on the comment icon inside the question card.

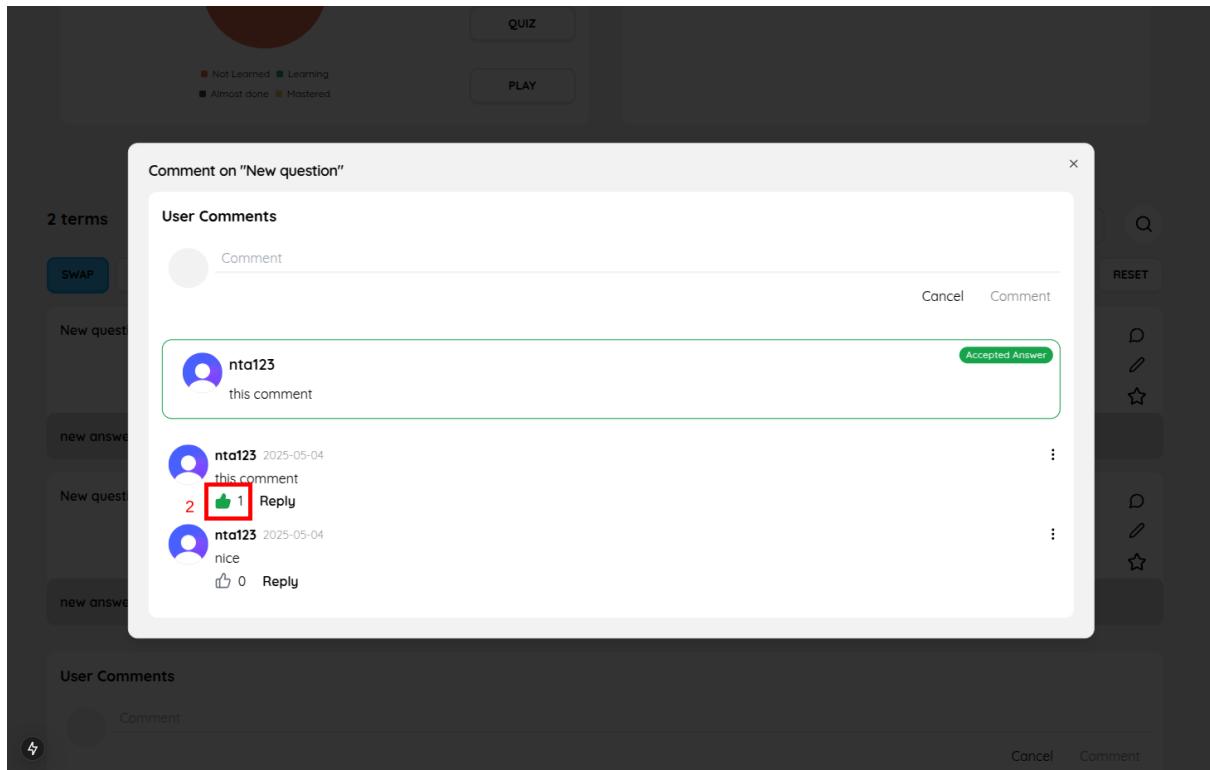


Figure 3.3.26.2.b: Upvote comments - 2

- (2) User clicks on the upvote icon inside the comment card to complete upvoting a comment.
- If a comment is already voted, clicking the icon again will un vote the comment

3.3.27 Mark question comments

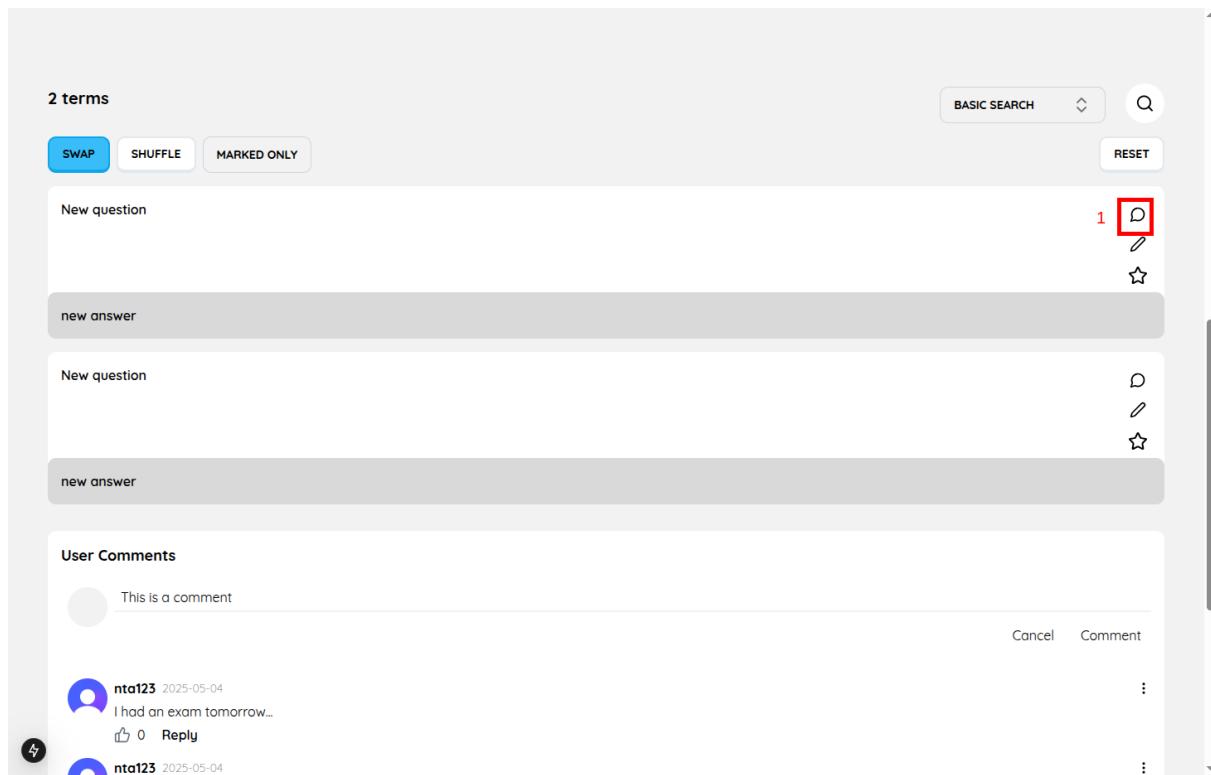


Figure 3.3.27.a: Mark question comments - 1

- (1) User clicks on the comment icon inside the question card.

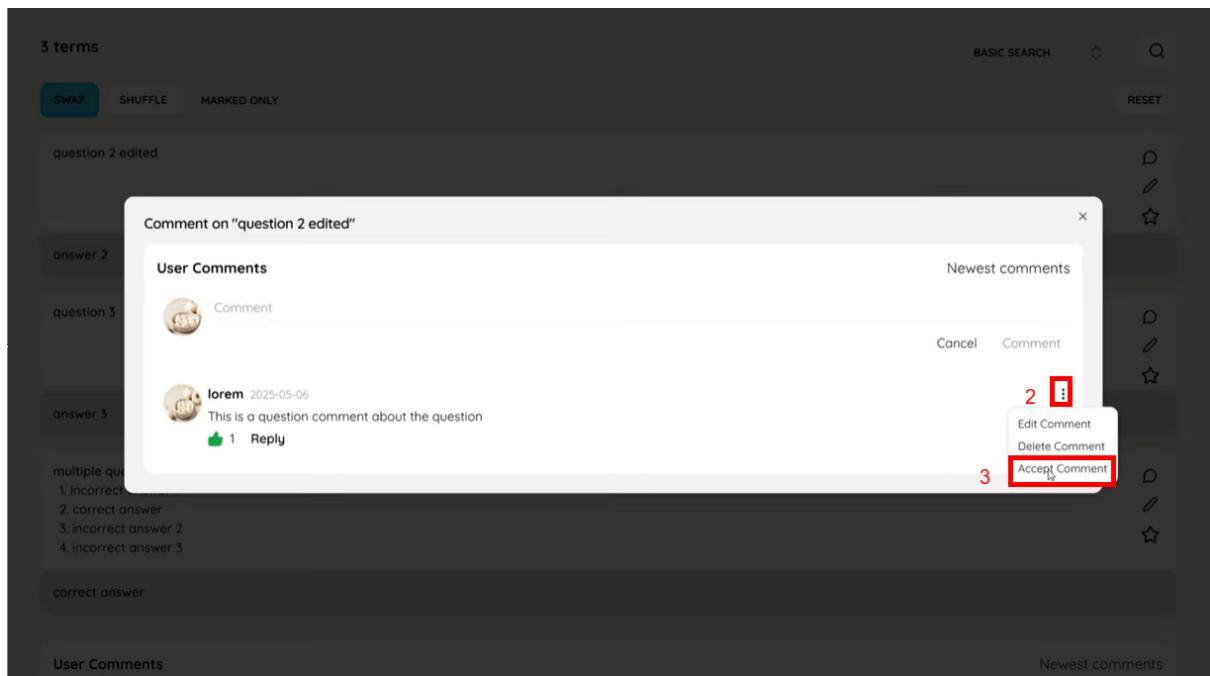


Figure 3.3.27.a: Mark question comments - 1

- (2) User clicks on the dropdown menu inside the comment card.
- (3) User clicks on the accept comment button to complete marking a comment
- Each question can only have one marked comment, if the user marks another comment, this will unmark the previously marked comment.

3.3.28 Learner and Guest filter comments

3.3.28.1 Filter question set comments

The screenshot shows a user interface for managing comments on a question set. At the top, there is a list of four terms: "2. A car moving on a highway", "3. A fan spinning", "4. Sound waves traveling". Below this is a search bar containing the text "A book on a high shelf". A button labeled "SHOW 5 MORE TERMS" is visible. On the right side, there is a star icon. In the bottom right corner of the main content area, a dropdown menu is open, showing three options: "1 Newest comments" (highlighted with a red box), "2 Newest comments" (also highlighted with a red box), "Top comments", and "Within Last Week".

User Comments

Comment

nta123 2025-05-05
This question might be too easy for the difficulty level
1 Reply

nta123 2025-05-05
This question might be too easy for the difficulty level
1 Reply

Quizzzee

Helping you master knowledge, one card at a time

Contact us

Home →
Explore →
Profile →

0123456789
quizzzee192@gmail.com

Figure 3.3.28.1: Filter comments

- (1) User clicks on the filter button inside the comment section.
- (2) User chooses an option to filter comments.

3.3.28.2 Filter question comments

The screenshot shows a user interface for filtering question comments. At the top, there is a search bar labeled "BASIC SEARCH" with a magnifying glass icon. To the right of the search bar are "RESET" and "SWAP" buttons. Below the search bar, there are two "New question" cards, each with a "new answer" button. In the middle of the screen, there is a "User Comments" section. It contains two comments from a user named "nta123" dated "2025-05-04". The first comment says "I had an exam tomorrow..." and has a reply count of 0. The second comment also has a reply count of 0. At the bottom right of the "User Comments" section, there is a "Comment" button. A red box highlights the comment icon in the "User Comments" section.

Figure 3.3.28.2.a: Filter comments - 1

- (1) User clicks on the comment icon inside the question card.

15 terms

BASIC SEARCH 🔍 RESET

SWAP SHUFFLE MARKED ONLY

Which of the following is the formula for kinetic energy?

1. $\frac{1}{2}mv^2$
2. mgh
3. $F=ma$
4. $P=IV$

$\frac{1}{2}mv^2$

Comment on "Which of the following is the formula for kinetic energy?"

User Comments

Comment

nta123 2025-05-05 This question might be too easy for the difficulty level

1 Reply

2 Newest comments

3 Newest comments

Top comments

Within Last Week

What is the value of the gravitational constant?

1. $3 \times 10^8 \text{ N m}^2 \text{ kg}^{-2}$
2. $5 \times 10^8 \text{ N m}^2 \text{ kg}^{-2}$
3. $3 \times 10^9 \text{ N m}^2 \text{ kg}^{-2}$
4. $3 \times 10^{10} \text{ N m}^2 \text{ kg}^{-2}$

$3 \times 10^8 \text{ m/s}^2$

Which law of motion states that for every action, there is an equal and opposite reaction?

1. Newton's Third Law
2. Newton's First Law
3. Newton's Second Law
4. Law of Conservation of Momentum

Newton's Third Law

What is the SI unit of electric current?

Figure 3.3.28.2.b: Filter comments - 2

- (2) User clicks on the filter icon inside the comment section.
- (3) User chooses an option to filter comments.

3.3.29 Learner and Guest use learn mode

The screenshot shows a user profile at the top left with the handle 'thegaelord' and a 'FOLLOW' button. Below the profile is a purple circular user icon. A message says 'I think this Quizzzee's rating should be 4 stars' with a 4-star rating icon underneath.

The main area features a 'Learning Progress' section with a large orange circle. To its right is a numbered list starting with '1 FLASH CARD', which is highlighted with a red box. Other items in the list are 'PRACTICE', 'QUIZ', and 'PLAY'. Below this is a legend: 'Not Learned' (orange square), 'Learning' (teal square), 'Almost done' (dark blue square), and 'Mastered' (yellow square).

Below the progress section, there is a list titled '15 terms' with three buttons: 'SWAP' (blue), 'SHUFFLE' (grey), and 'MARKED ONLY' (grey). A question is displayed: 'Which of the following is the formula for kinetic energy?'. Four options are listed: 1. $\frac{1}{2}mv^2$, 2. mgh , 3. $F=ma$, and 4. $P=IV$. To the right of the question are icons for 'DRAFT' (circle with a lightning bolt) and 'STAR' (star).

At the top right of the interface are 'BASIC SEARCH' and 'RESET' buttons, along with a magnifying glass icon.

Figure 3.3.29.a: Use learn mode - 1

- (1) User clicks on the Learn button to be redirected to the learn page.

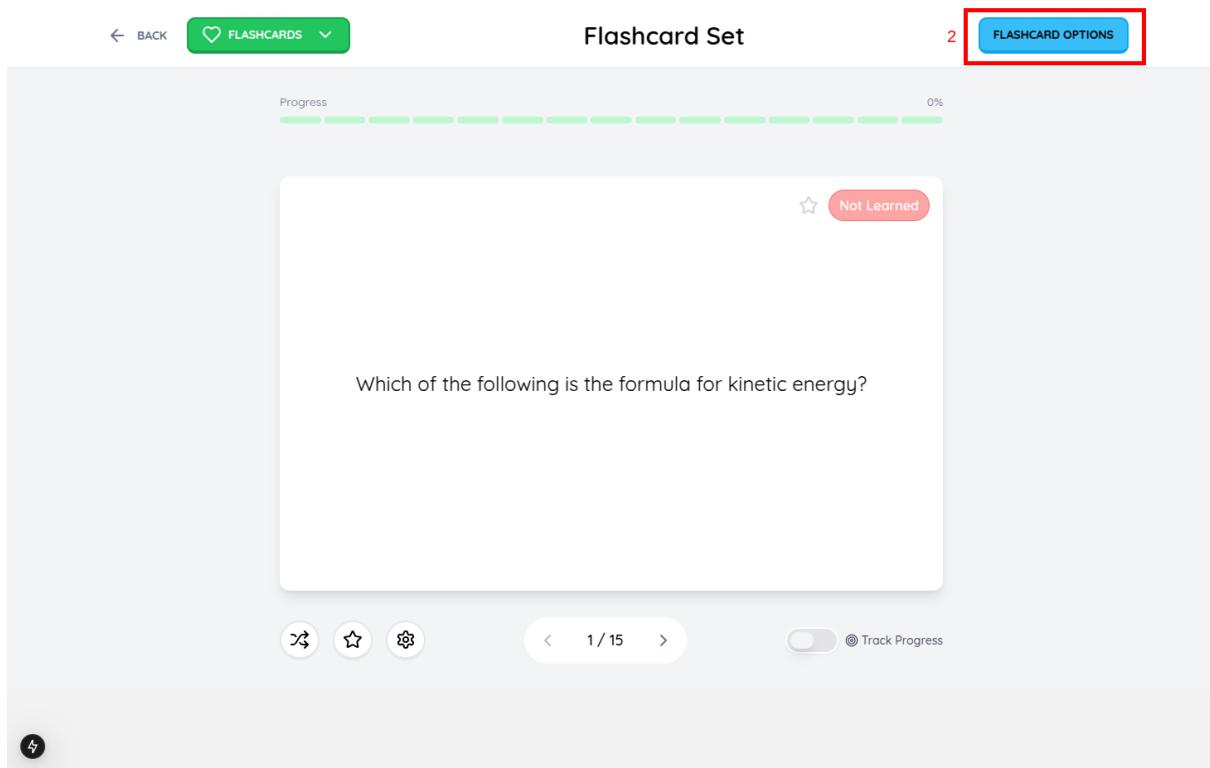


Figure 3.3.29.b: Use learn mode - 2

- (2) User can customize the Learning process by clicking the option button.

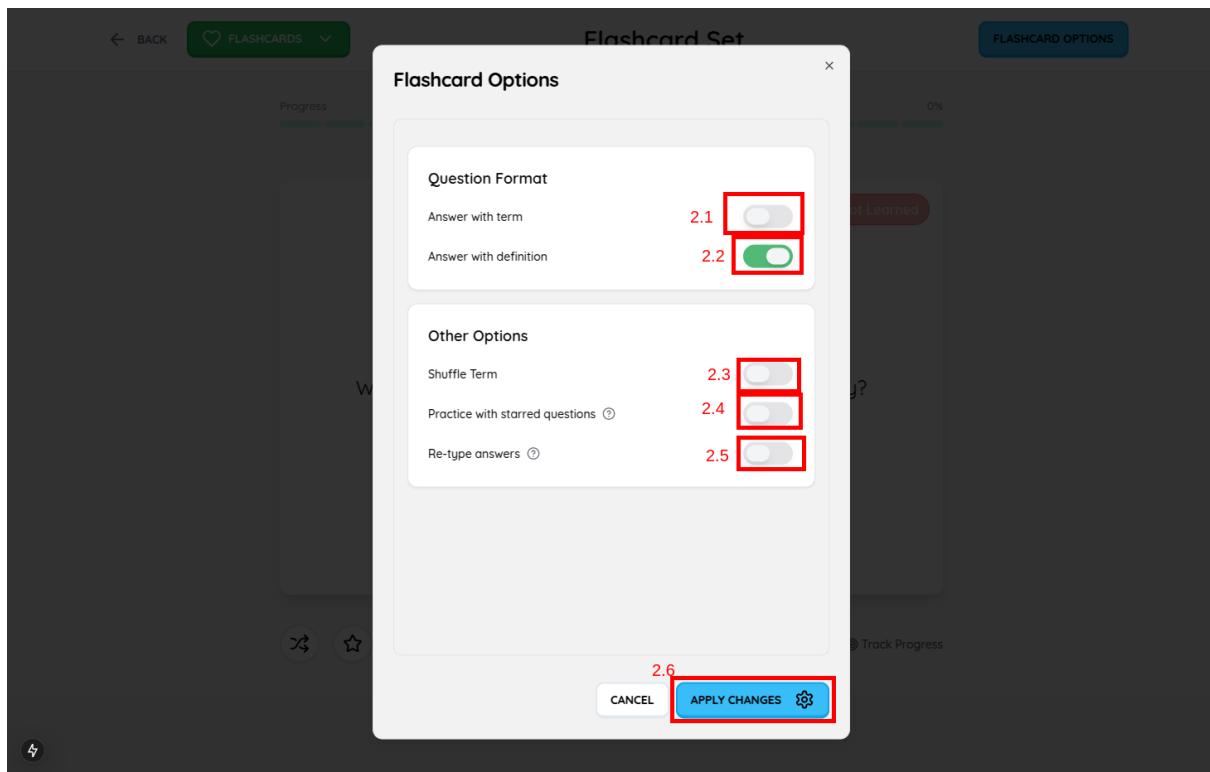


Figure 3.3.29.c: Use learn mode - 3

- (2.1) User chooses the answer with term switch.
- (2.2) User chooses the answer with definition switch.
- (2.3) User chooses to shuffle terms
- (2.4) User chooses to practice with marked question.
- (2.5) User chooses to re-type answers
- (2.6) User clicks apply change to complete customization.

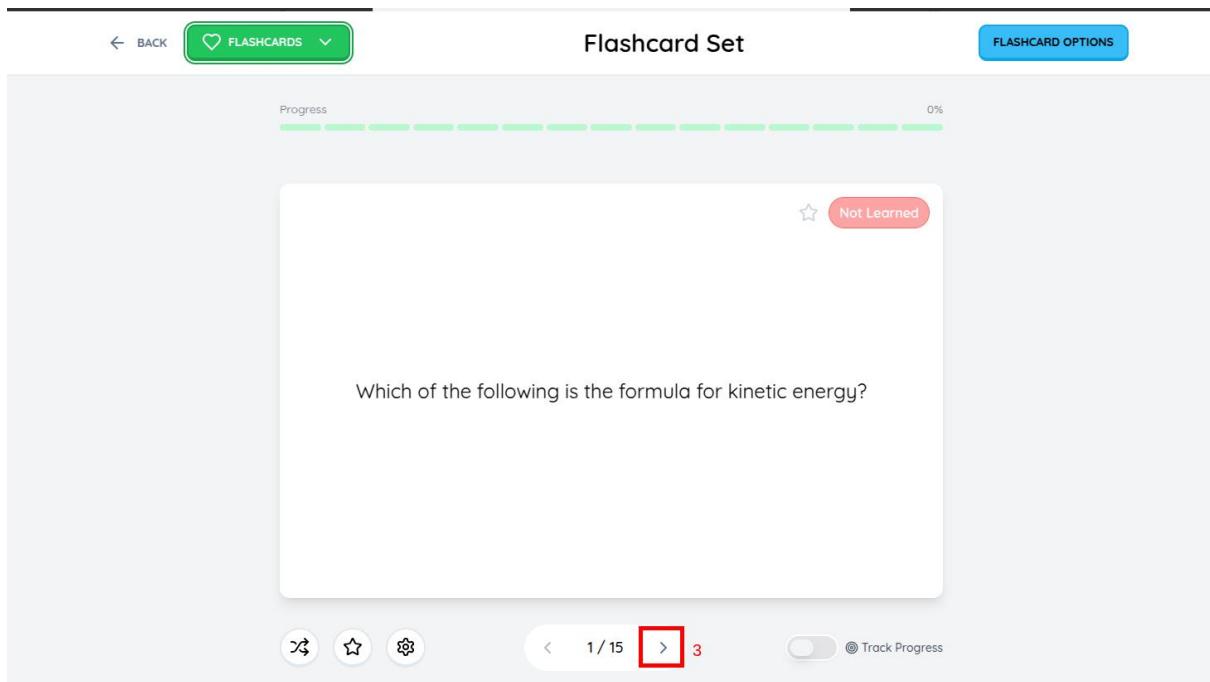


Figure 3.3.29.d: Use learn mode - 4

- (3) User clicks on the next button to goes to the next question

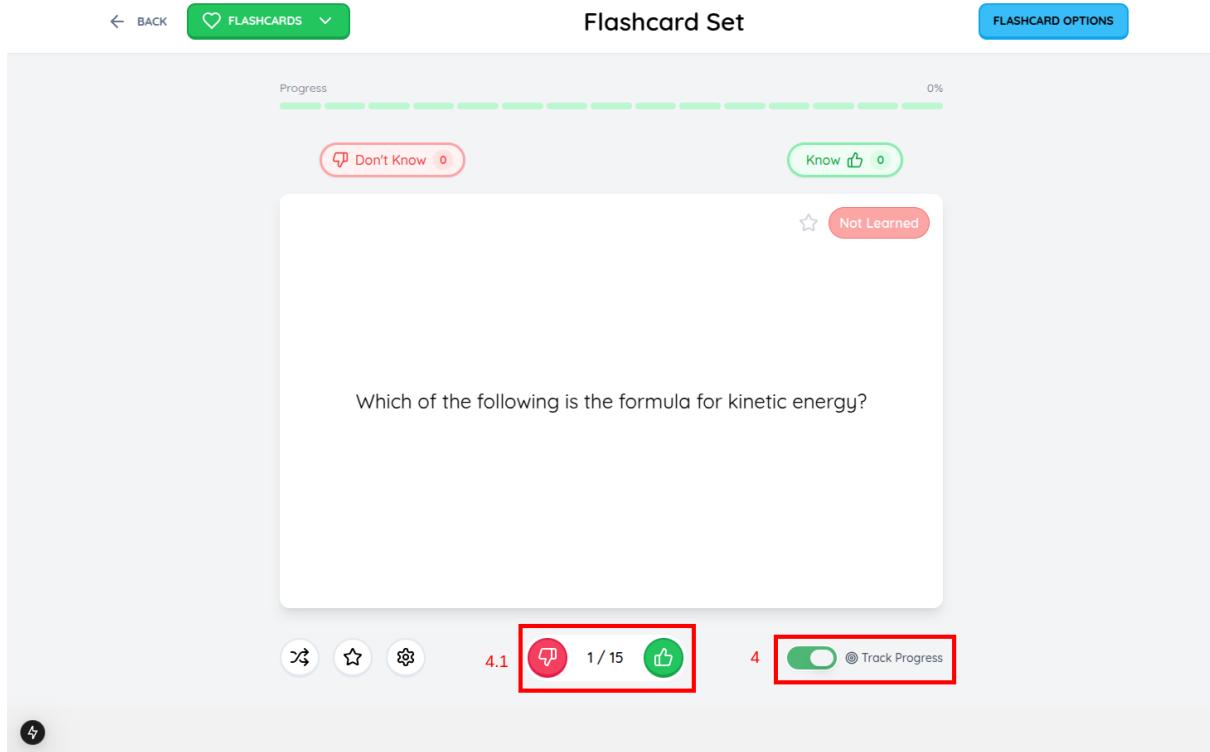


Figure 3.3.29.e: Use learn mode - 5

- (4) User toggle track progress to start tracking their question set knowledge (optional)
- Tracking only shows which question the user has known, no progress will be saved.
- (4.1) User clicks on the thumb up or thumb down.

3.3.30 Learner and Guest use matching mode

The screenshot shows a user profile for 'thegaelord' with a blue and red icon. Below the profile, there is a message from another user: 'I think this Quizzzee's rating should be 4 stars' followed by five yellow stars. A 'Learning Progress' section contains a pie chart divided into four segments: Not Learned (red), Learning (dark green), Almost done (dark blue), and Mastered (yellow). The 'Mastered' segment is the largest. To the right of the chart are buttons for 'FLASH CARD', 'PRACTICE', 'QUIZ', and 'PLAY'. The 'PLAY' button is highlighted with a red rectangle. At the bottom left, it says '15 terms' with buttons for 'SWAP', 'SHUFFLE', and 'MARKED ONLY'. On the right side, there are 'BASIC SEARCH' and 'RESET' buttons.

Figure 3.3.30.a: Use matching mode - 1

- (1) User clicks on the play button to be redirected to the matching page.

HOME EXPLORE Search with keyword

EN 50

Which particle has a positive charge?

Which of these is a vector quantity?

Velocity

What is the SI unit of electric current?

What is the speed of light in vacuum?

Ampere

3×10^8 m/s

Proton

Figure 3.3.30.b: Use matching mode - 2

- User clicks on each pair of cards until all the card pairs have been lifted.

3.3.31 Learner and Guest Use quiz mode

The screenshot shows a user profile page for 'thegaelord'. At the top, there are several green circular icons followed by a dropdown menu. Below this, the word 'Tags:' is followed by three colored boxes: orange ('Classical Mechanics'), green ('Quantum Physics'), and purple ('Electromagnetism'). A heart icon is in the top right corner.

Under the profile picture, the username 'thegaelord' is displayed next to a 'FOLLOW' button. A blue circular placeholder for a profile picture is shown below the main one.

A message from another user says: 'I think this Quizzzee's rating should be 4 stars' with a rating of ★★★★☆.

The central area features a 'Learning Progress' pie chart divided into four segments: Not Learned (red), Learning (dark teal), Almost done (dark blue), and Mastered (yellow). A legend below the chart defines these colors. To the right of the chart are four buttons: 'FLASH CARD', 'PRACTICE', 'QUIZ' (which is highlighted with a red border and has a red number '1' above it), and 'PLAY'.

Below this section, there is a large empty white box.

At the bottom left, it says '15 terms' and has buttons for 'SWAP' (with a lightning bolt icon), 'SHUFFLE', and 'MARKED ONLY'. On the right side, there is a 'BASIC SEARCH' input field with a magnifying glass icon, a 'RESET' button, and a small downward arrow.

Figure 3.3.31.a: Use quiz mode - 1

- (1) User clicks on the Quiz button to be redirected to the quiz page.

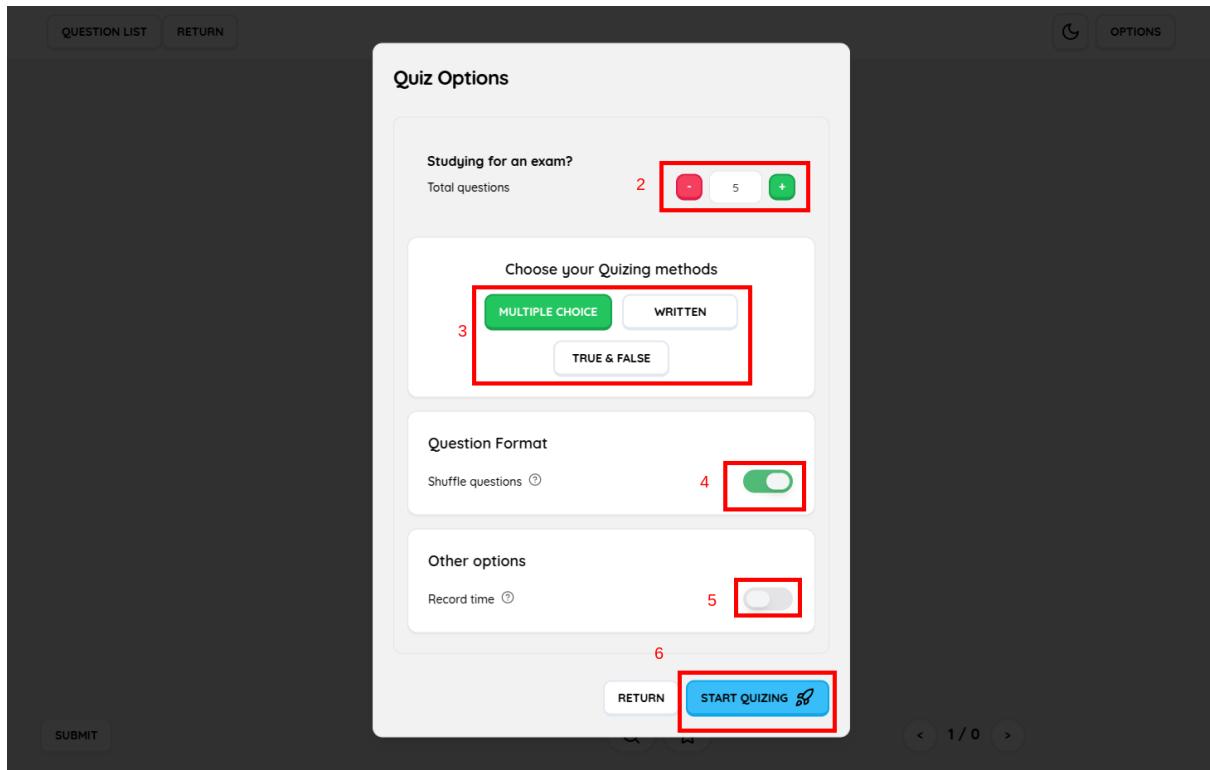


Figure 3.3.31.b: Use quiz mode - 2

- Inside the option modal, the user customizes their quiz.
- (2) User changes the number of questions.
- (3) User chooses a method to do a quiz.
- (4) Use enables shuffling the set (optional)
- (5) Use enables record time (optional)
- (6) User clicks the start button to begin the quiz.

The screenshot shows a quiz interface. At the top left, there is a sidebar with a red box around the number '8.1' and the text 'QUESTION LIST'. To its right are 'RETURN' and 'OPTIONS' buttons. On the far right are a refresh icon and another 'OPTIONS' button. Below the sidebar is a large empty white area. To the right of this area are three boxes: 'Watt' (empty), 'Volt' (empty), and 'Ampere' (highlighted with a red box). Below these boxes is an empty box labeled 'Ohm'. At the bottom left is a 'SUBMIT' button. In the center bottom are search and star icons. At the bottom right is a navigation bar with arrows, the text '1 / 5', and a red box around the number '8'.

What is the SI unit of electric current?

7

Ampere

Watt

Volt

Ohm

SUBMIT

1 / 5

8

Figure 3.3.31.c: Use quiz mode - 3

- (7) User answers the question.
- (8) User clicks the next button to go to the next question.
- (8.1) The User can instantly go to one of the questions by clicking the title from the sidebar.

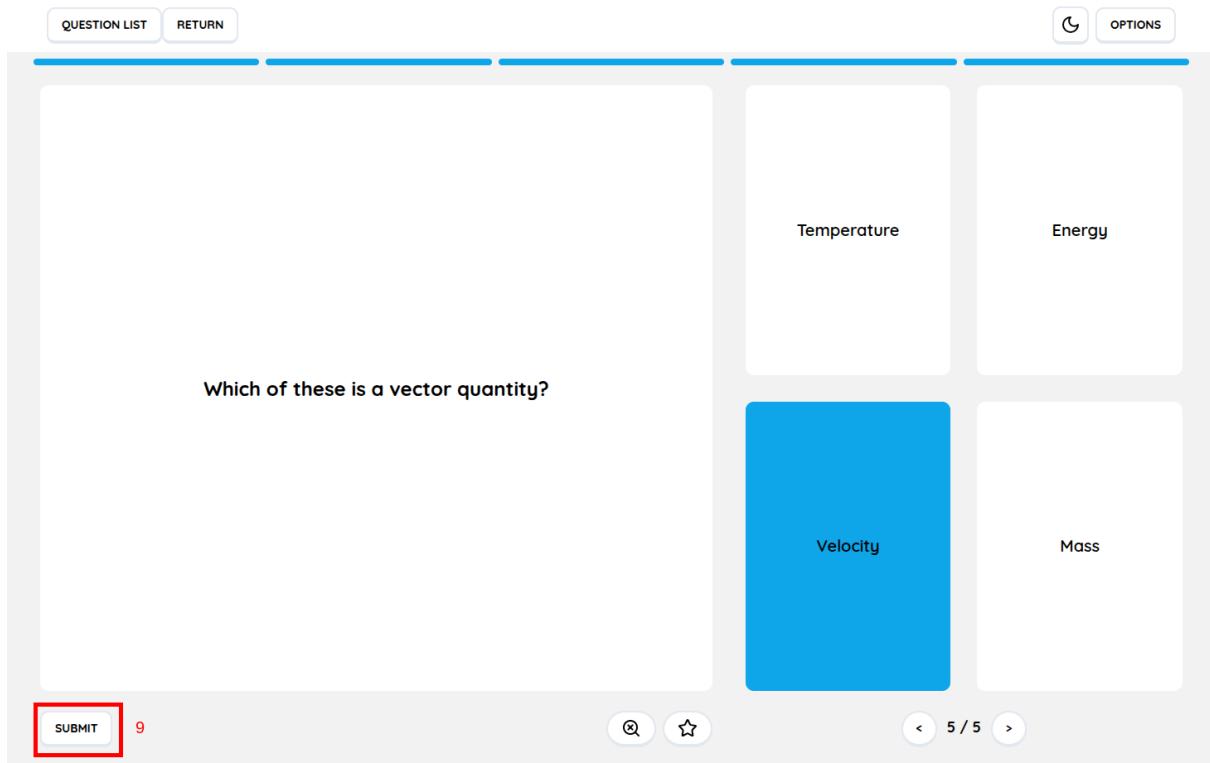


Figure 3.3.31.d: Use quiz mode - 4

- (9) Once finished, the user clicks the submit button to finish the quiz.

3.3.32 Learner view quiz statistics

The screenshot shows a user profile for 'thegaelord' with a blue circular icon. Below the profile, a message says 'I think this Quizzzee's rating should be 4 stars' with a 4-star rating icon. A 'Learning Progress' section features a pie chart divided into four segments: Not Learned (red), Learning (teal), Almost done (dark blue), and Mastered (yellow). Buttons for 'FLASH CARD', 'PRACTICE', 'QUIZ', and 'PLAY' are available. To the right, a red box highlights a list of four quiz attempts from May 5, 2025:

Date	Score	Action
2025-05-05	Score: 4	REVISIT
2025-05-05	Score: 5	REVISIT
2025-05-05	Score: 1	REVISIT
2025-05-05	Score: 3	REVISIT

At the bottom left, there are buttons for 'SWAP', 'SHUFFLE', and 'MARKED ONLY'. On the right, there are 'BASIC SEARCH' and 'RESET' buttons.

Figure 3.3.32: View quiz statistics

- Inside the question set detail page, the user will find charts that store user quiz attempt statistics, including the learning status of questions.
- This will only appear once the learner has done a quiz.

3.3.33 Learner revisit quiz attempt

The screenshot shows a user profile for 'thegaelord' with a blue circular icon. Below the profile, there's a placeholder for another user's profile with a blue circular icon and the text 'I think this Quizzzee's rating should be 4 stars' followed by a 4-star rating icon.

Learning Progress: A pie chart divided into four segments: Not Learned (red), Learning (teal), Almost done (dark blue), and Mastered (yellow). The legend below the chart indicates: Not Learned (red), Learning (teal), Almost done (dark blue), and Mastered (yellow).

Flashcard Options: Buttons for FLASH CARD, PRACTICE, QUIZ, and PLAY.

Quiz History: A table showing quiz attempts:

Date	Score	Action
2025-05-05	Score: 4	REVISIT (button highlighted with red box)
2025-05-05	Score: 5	REVISIT
2025-05-05	Score: 1	REVISIT
2025-05-05	Score: 3	REVISIT

Bottom Controls: Buttons for SWAP, SHUFFLE, MARKED ONLY, BASIC SEARCH, Q (search icon), and RESET.

Figure 3.3.33.a: Revisit quiz attempt - 1

- Under the quiz statistics, the user can find the list of quiz attempts they have taken
- (1) User clicks on the revisit button to be redirected to the quiz page.

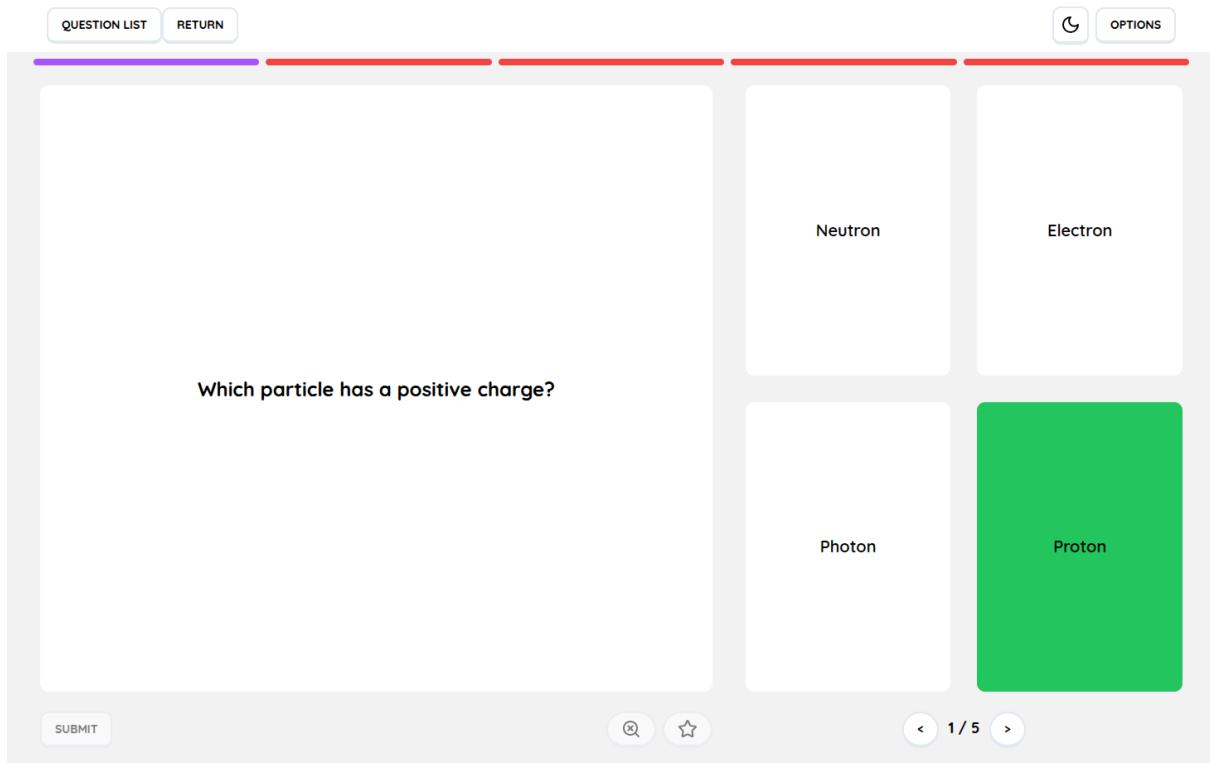


Figure 3.3.33.a: Revisit quiz attempt - 1

- Inside the page, the user will see the previous attempt they have made.

3.3.34 View groups

3.3.34.1 Learner and Guest view groups when search

The screenshot shows a web-based learning platform interface. At the top, there is a navigation bar with icons for HOME, EXPLORE, and a search bar containing the placeholder "Search with keyword". A red box highlights the search bar. To the right of the search bar are various user and system status indicators, including a notification count of 1. Below the navigation bar, a sidebar titled "Pick up where you left" contains a list of "Test Question Set #2". This set is described as having 15 questions and being "Halfway done". It includes a user icon for "thegaelord", a progress bar at 67%, and categories: Classical Mechanics, Quantum Physics, and Electromagnetism. A rating of 4.0 stars is shown. Below this, there is a quote by Mahatma Gandhi: "Live as if you were to die tomorrow. Learn as if you were to live forever." attributed to him. At the bottom of the sidebar, under the heading "Data Structures and Algorithms", are links for "Test Question Set #4" and "Test Question Set #8". A red box highlights the search bar.

Figure 3.3.34.1.a: View groups - 1

- (1) User clicks on the search bar

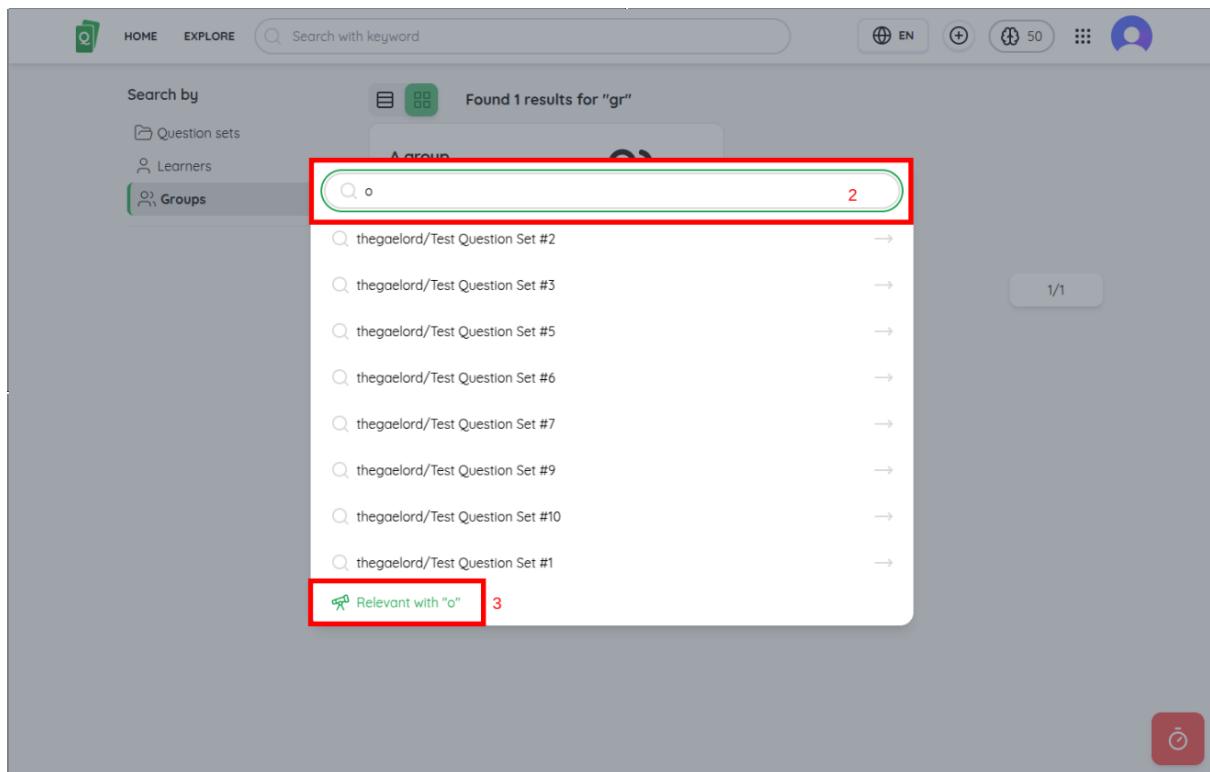


Figure 3.3.34.1.b: View groups - 2

- (2) User enter a keyword inside the prompt
- (3) User presses the enter key or clicks the “relevant” link to be redirected to the search result page.

The screenshot shows a user interface for searching. At the top, there are navigation links for 'HOME' and 'EXPLORE', a search bar with placeholder text 'Search with keyword', and various status indicators including language ('EN'), notifications ('50'), and a user profile icon. Below the search bar, a sidebar titled 'Search by' lists categories: 'Question sets', 'Learners', and 'Groups'. The 'Groups' option is highlighted with a red box and has a red number '4' next to it. The main content area displays search results for the query 'o'. A summary box indicates 'Found 8 results for "o"' and shows a group named 'A group' with a profile picture of two people, '1 sets', and '1 members'. The bottom right corner features a red circular button with a white icon.

Figure 3.3.34.1.c: View groups - 3

- (4) Inside the search result page, user filter for group to complete viewing groups.

3.3.34.2 Learner view participated groups

The screenshot shows a user's profile at the top right, including a green icon, a name, a '50' badge with a red notification dot, and a 'My Groups' button. A red box highlights the 'My Groups' button. A dropdown menu is open from this button, listing 'My Sets', 'Goals', and 'Achievements'. Below the profile, a section titled 'Pick up where you left' displays a 'Test Question Set #2' by 'thegaelord' with 15 questions, a progress bar at 67% (Halfway done), and categories like 'Classical Mechanics', 'Quantum Physics', and 'Electromagnetism'. A quote by Benjamin Franklin is shown in a box: "Tell me and I forget. Teach me and I remember. Involve me and I learn." - Benjamin Franklin. At the bottom, there are two more sections: 'Data Structures and Algorithms' and 'Test Question Set #4'.

Figure 3.3.34.2.a: View groups - 1

- (1) User clicks the more button
- (2) User clicks my groups button to be redirected to the groups page.

The screenshot shows the 'Your Groups' section of a web application. At the top, there are navigation links for 'HOME' and 'EXPLORE', a search bar labeled 'Search with keyword', and several user profile and settings icons. Below the header, the title 'Your Groups' is displayed, followed by the subtitle 'Manage your study groups'. A large, prominent button labeled 'Groups' is highlighted. To the right of the main content area, there is a search bar with the placeholder 'Search groups...' and a button labeled 'INVITE'. The main content area displays a single group entry: 'A group' (with a person icon), '1 sets', and '1 members' (with the note 'You are the Leader'). Below this, a small '1/1' indicator is visible. The bottom right corner features a red circular icon with a white symbol.

Figure 3.3.34.2.b: View groups - 2

- Inside the group page, the user can see a list of their participating groups.

3.3.34.3 The Administrators view groups inside the groups page.

The screenshot shows the Quizzzee application interface. On the left, there is a sidebar with the following navigation options: Dashboard, Learners, Question sets, Comments, Groups (which is highlighted with a red box), Reports, Subjects, and Fields. The main content area is titled "Groups" and displays a table with one row. The table columns are: Group Name, Description, Leader, Public, Active, Created At, and Updated At. The single row contains the values: A group, A description, nta123, Yes, Yes, May 4, 2025, and May 4, 2025. At the bottom right of the table, there are "PREVIOUS" and "NEXT" buttons.

Group Name	Description	Leader	Public	Active	Created At	Updated At
A group	A description	nta123	Yes	Yes	May 4, 2025	May 4, 2025

Figure 3.3.34.3 View groups

- User clicks on the groups button to be redirected to the groups page.
- Inside the groups page, the user can see the table of groups.

3.3.35 View group detail

3.3.35.1 Learner and guest view group detail

The screenshot shows a web interface for managing groups. At the top, there are navigation links for 'HOME' and 'EXPLORE', a search bar, and various user and system status indicators. Below the header, the title 'A group' is displayed, followed by a horizontal menu with tabs: 'Shared Sets' (which is underlined), 'Members', 'Description', 'Join Requests', and 'Banned'. To the right of the menu is a 'Settings' link. Underneath the menu, it says 'Current Mode: Shared Sets' with buttons for 'RECENT', 'ALL', and '+ ADD SET'. A search bar for sets is also present. A date indicator 'May 2025' is shown above a list of items. The first item in the list is a card for 'Test Question Set #2', which includes a user icon and a small red button in the bottom right corner.

Figure 3.3.35.1: View group detail

- User clicks the group card to be redirected to the group detail page.
- Inside the group detail page, the user can view the details of a group.

3.3.35.2 The Administrator views group detail

The screenshot shows the Quizzzee application interface. On the left is a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. The main content area has a header "Group Testing" with a subtitle "Math please" and a note "Created on May 3, 2025". A red "BAN GROUP" button is in the top right. Below the header are two sections: "Group Settings" and "Group Stats". "Group Settings" includes fields for "Public" (Yes), "Can Invite" (Yes), and "Can Add Question Sets" (Yes). "Group Stats" shows 1 member and 1 question set. A "Group Leader" section shows a profile picture of a user named Tory Tanner. The bottom left corner shows a user profile for "torynothing".

Figure 3.3.35.2 View group detail

- User clicks on the group table row to be redirected to the group detail page.

3.3.36 Learner creates group

The screenshot shows a user interface for managing study sets or groups. At the top, there are navigation links for 'HOME' and 'EXPLORE', a search bar, and a user profile icon. A red box highlights the 'Create set' button. Another red box highlights the 'Create group' button, which is located next to it. Below the header, there are three study set cards:

- A title**: Created by nta123, 2 questions. Description: A description. Tags: General Biology. Rating: 0 stars.
- This title**: Created by nta123, 2 questions. Description: this description. Tags: General Biology, Electromagnetism. Rating: 0 stars.
- Test Question Set #2**: Created by thegaelord, 15 questions. Description: Understand the basic principles of physics, from motion to energy. Tags: Classical Mechanics, Quantum Physics, Electromagnetism. Rating: 4.0 stars.

At the bottom of the screen, there is a quote: "Education is the most powerful weapon which you can use to change the world."

Figure 3.3.36.a: Create groups - 1

- (1) User clicks on the plus icon button.
- (2) User clicks on the create group button to be redirected to the create group page.

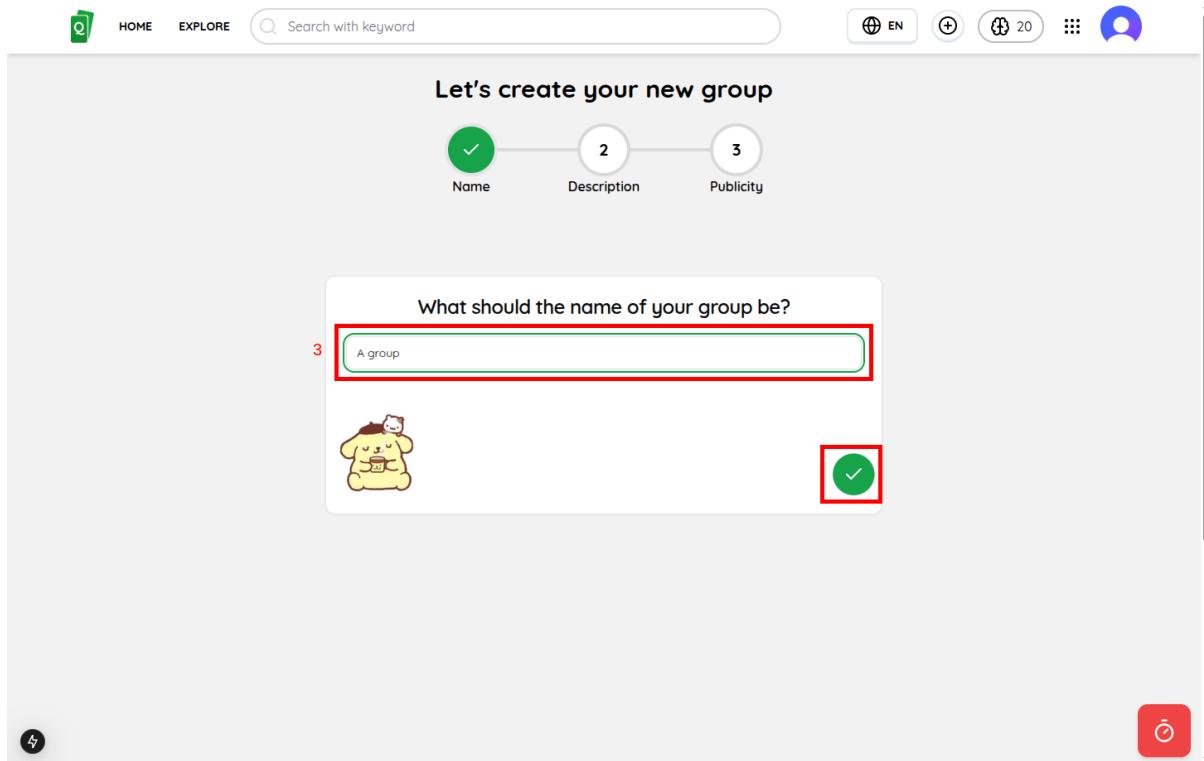


Figure 3.3.36.b: Create groups - 2

- (3) User inputs a group name

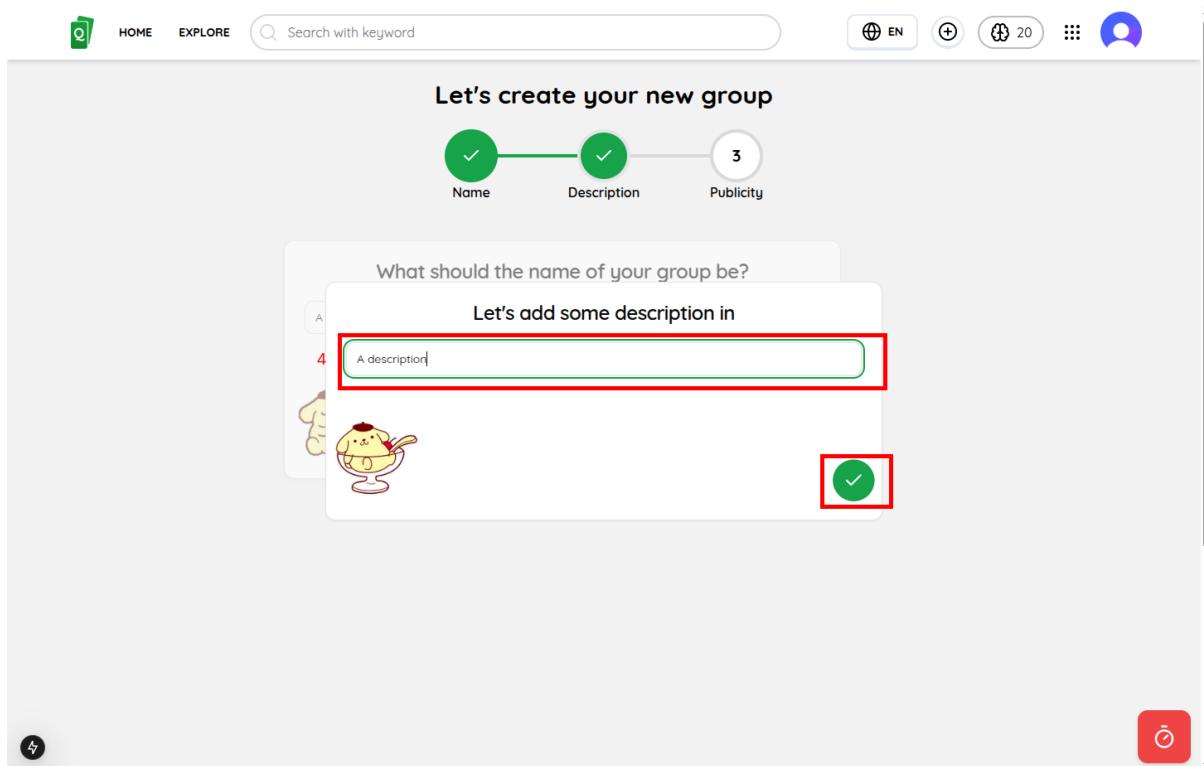


Figure 3.3.36.c: Create groups - 3

- (4) User input a description.

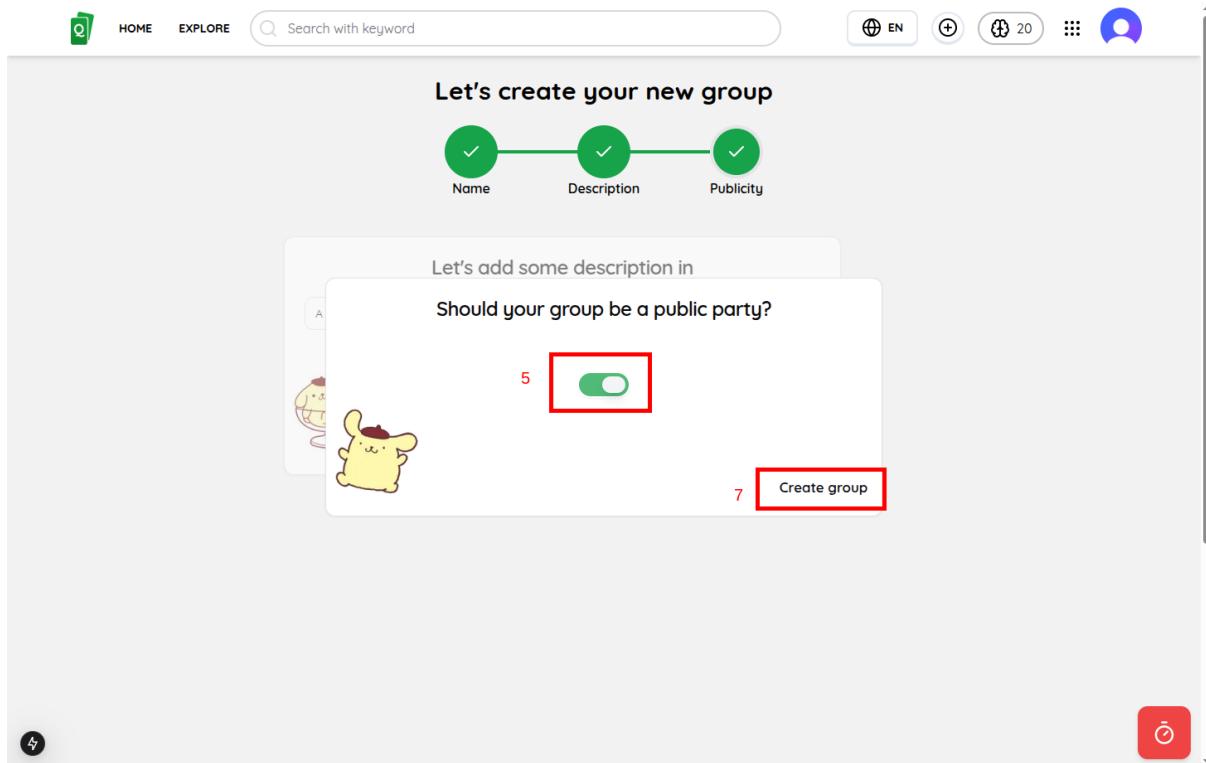


Figure 3.3.36.d: Create groups - 4

- (5) User set group access to the public. (optional)
- (6) User clicks the create button to complete creating a new group.

3.3.37 Learner update groups

3.3.37.1 Update group settings

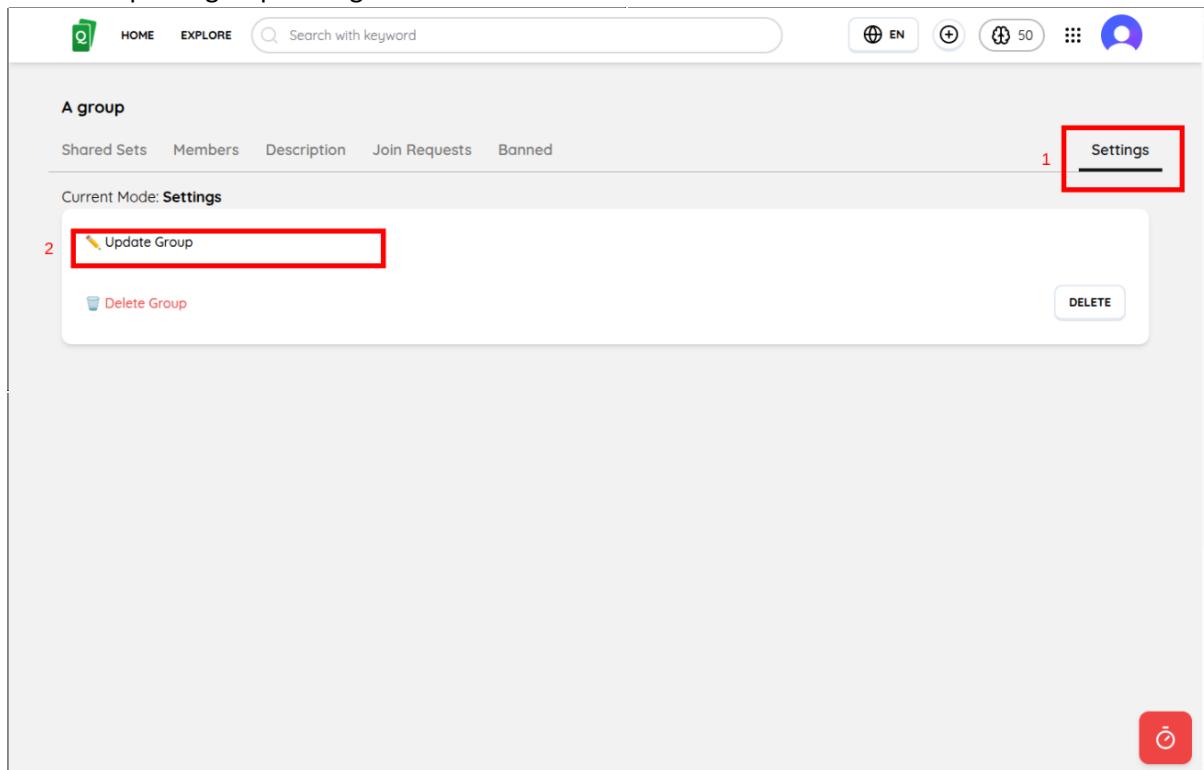


Figure 3.3.37.a: Update groups - 1

- (1) Inside the group detail page, user clicks on the settings button.
- (2) User clicks on the update collapsible

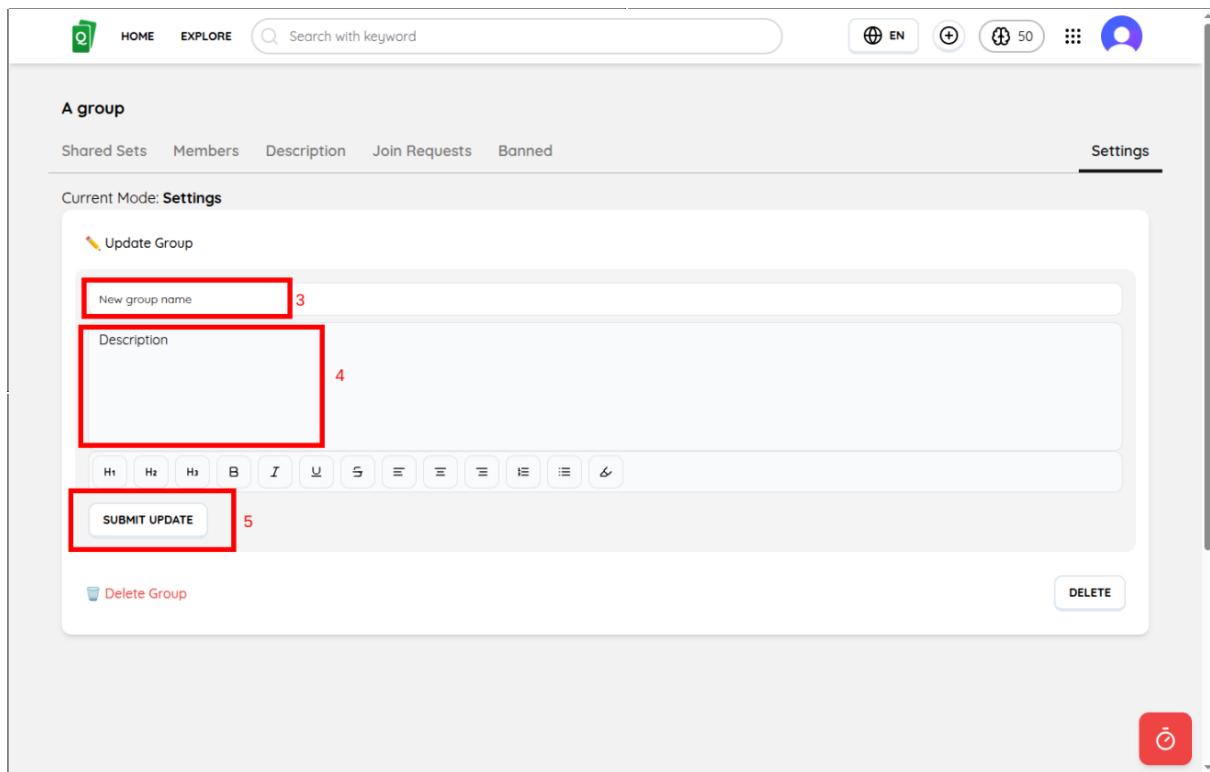


Figure 3.3.37.a: Update groups - 1

- (3) User inputs a title
- (4) User inputs a description
- (5) User clicks the update button to complete updating their group.

3.3.37.2 Remove member

The screenshot shows a user interface for managing group members. At the top, there's a navigation bar with icons for HOME, EXPLORE, and a search bar. Below the navigation is a header for 'Math group'. The 'Members' tab is highlighted with a red box and labeled '1'. Underneath, it says 'Current Mode: Members' with a 'JOIN DATE' dropdown and a '+' button. To the right is a search bar with 'Search members...'. The main area displays two member cards. The first member, 'Tory Tanner', has a pink profile picture with a white 'T', the handle '@oooooooo', and 1 follower. The second member, 'Lorem Issump', has a profile picture of a bowl of garlic, the handle '@lorem', and 1 follower. Between the two cards is a small red box labeled '2' with a menu icon. A larger red box labeled '3' surrounds the 'Kick member' option in the menu, which is also highlighted with a red box.

Figure 3.3.37.2 Update group

- (1) Inside the group detail page, the user clicks on the member tab
- (2) User clicks on the menu button inside the member card.
- (3) User clicks on the kick button.

3.3.37.3 Ban member

The screenshot shows a user interface for managing group members. At the top, there are navigation links: HOME, EXPLORE, and a search bar labeled 'Search with keyword'. On the right side, there are icons for language (EN), notifications (20), and profile settings.

The main area is titled 'Math group' and has tabs for 'Shared Sets', 'Members' (which is highlighted with a red box and labeled '1'), 'Description', 'Join Requests', and 'Banned'. Below this, it says 'Current Mode: Members' and shows a dropdown for 'JOIN DATE'.

The 'Members' section lists two users:

- Tory Tanner (@aaaaaaaaaa) with 1 follower. A green button below her profile picture says '1 follower'.
- Lorem Issump (@lorem) with 1 follower. A green button below his profile picture says '1 follower'.

For the user 'Tory Tanner', a context menu is open, indicated by a red box labeled '2' over the menu icon. The menu options are 'Change leader' and 'Ban member'. The 'Ban member' option is also highlighted with a red box and labeled '3'.

Figure 3.3.37.3 Update group

- (1) Inside the group detail page, the user clicks on the member tab
- (2) User clicks on the menu button inside the member card.
- (3) User clicks on the ban button.

3.3.37.4 Change leader

The screenshot shows a group detail page titled "Math group". The "Members" tab is highlighted with a red box and labeled "1". The "Current Mode: Members" section includes "JOIN DATE" dropdowns and a search bar. Below, two member cards are shown: "Tory Tanner" (1 follower) and "Lorem Issump" (1 follower). A red box labeled "2" highlights the three-dot menu icon in the "Lorem Issump" card. A red box labeled "3" highlights the "Change leader" option in the resulting dropdown menu.

Figure 3.3.37.4 Update group

- (1) Inside the group detail page, the user clicks on the member tab.
- (2) User clicks on the menu button inside the member card.
- (3) User clicks on the change leader button.

3.3.38 Learner deletes groups

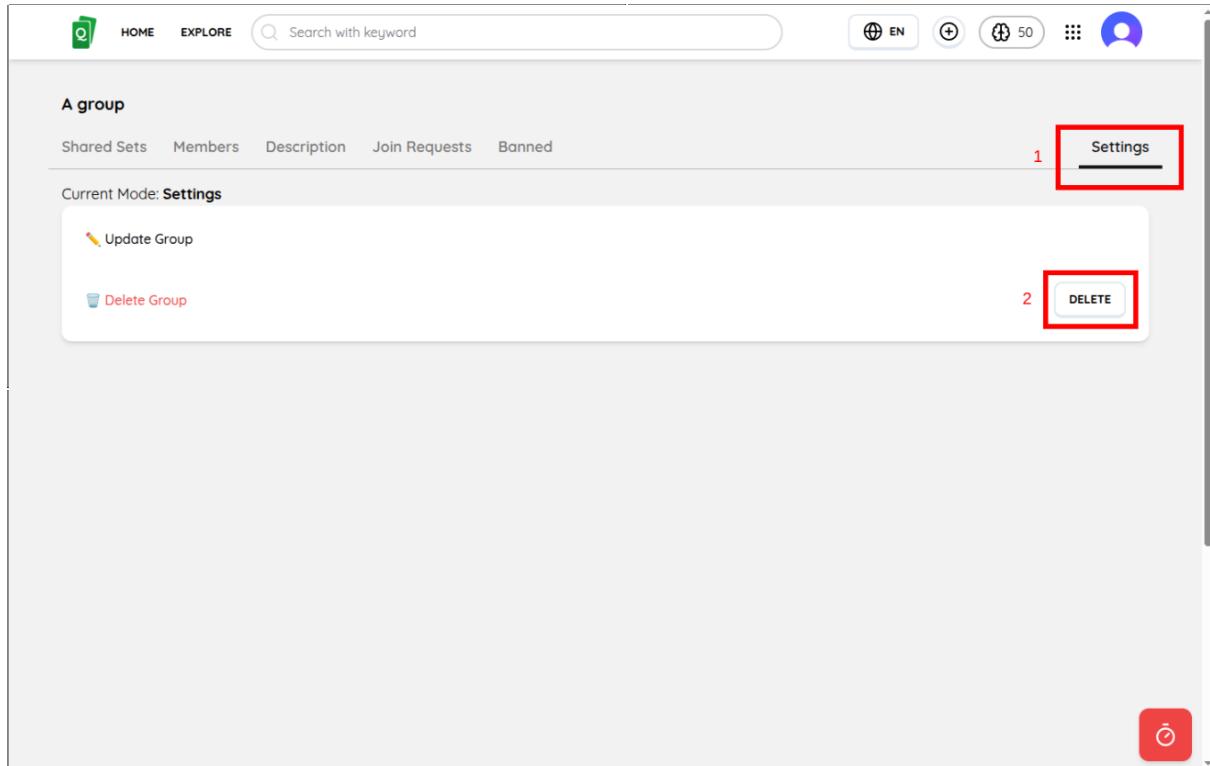


Figure 3.3.38.a Delete groups - 1

- (1) Inside the group detail page, the user clicks on the settings button.
- (2) User clicks on the delete button

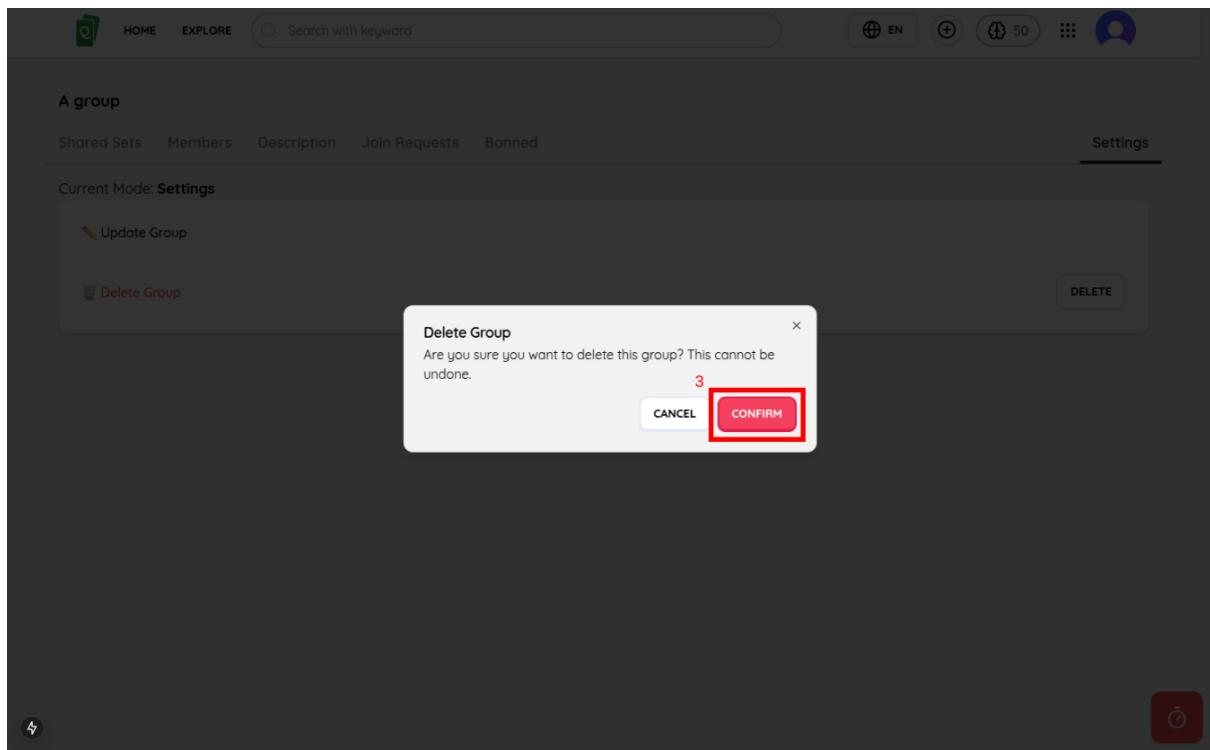


Figure 3.3.38.b Delete groups - 2

- (3) User clicks on the confirm button to complete deleting the group.

3.3.39 Learner sends invitations

The screenshot shows a user interface for managing study groups. At the top, there are navigation links for 'HOME' and 'EXPLORE', a search bar, and various user settings. Below this, the title 'Your Groups' is displayed, with the subtitle 'Manage your study groups'. There are two tabs: 'Groups' (which is selected) and 'Invitations'. A search bar for groups is also present. In the main area, a group named 'A group' is listed, showing '2 sets' and '1 member'. The user is identified as 'You are the Leader'. On the far right of the screen, there is a red box highlighting the 'INVITE' button, which has a red number '1' above it, indicating one invitation pending.

Figure 3.3.39.a: Send invitations - 1

- (1) Inside the groups page, the user clicks on the send invitation button.

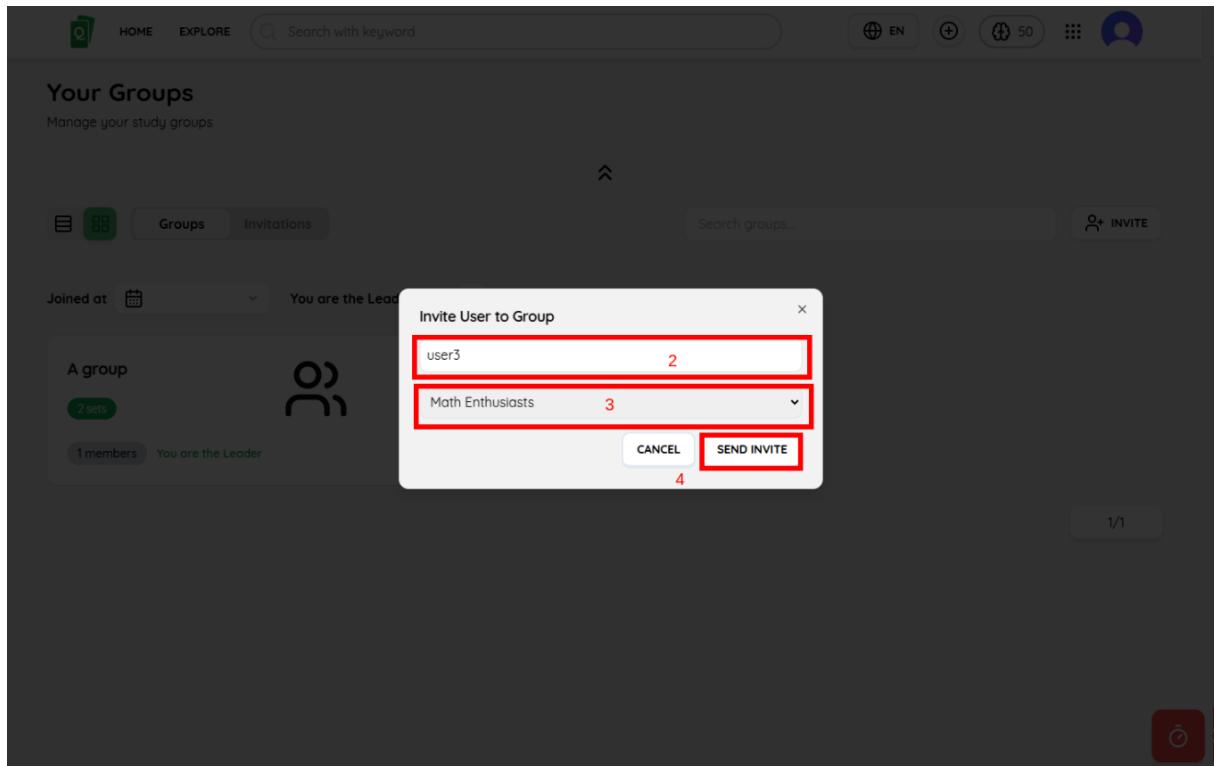


Figure 3.3.39.b: Send invitations - 2

- (2) User inputs a username.
- (3) User chooses a group.
- (4) User clicks the send invite button to complete sending an invitation.

3.3.40 Learner joins groups

The screenshot shows a user interface for managing study groups. At the top, there are navigation links for 'HOME' and 'EXPLORE', a search bar with placeholder 'Search with keyword', and various user account icons. Below this, a section titled 'Your Groups' is displayed with the sub-instruction 'Manage your study groups'. A horizontal navigation bar contains three items: 'Groups' (disabled), 'Invitations' (highlighted with a red box and the number '1'), and 'Search groups...' with a magnifying glass icon. To the right of this bar is a 'INVITE' button with a plus sign and a person icon. The main content area shows a card for an invitation to a group named 'A group', which was invited by 'user3' and expired on '20/03/2025'. Two buttons are present: 'Accept' (highlighted with a red box and the number '2') and 'Decline'. A red arrow points from the 'Accept' button towards the bottom of the page.

Figure 3.3.40 Join groups

- (1) Inside the groups page, the user clicks the invitation tab.
- (2) User clicks the accept button inside the invitation card to complete joining a group.
- The user can click the decline button to decline the invitation.

3.3.41 Learner leaves groups

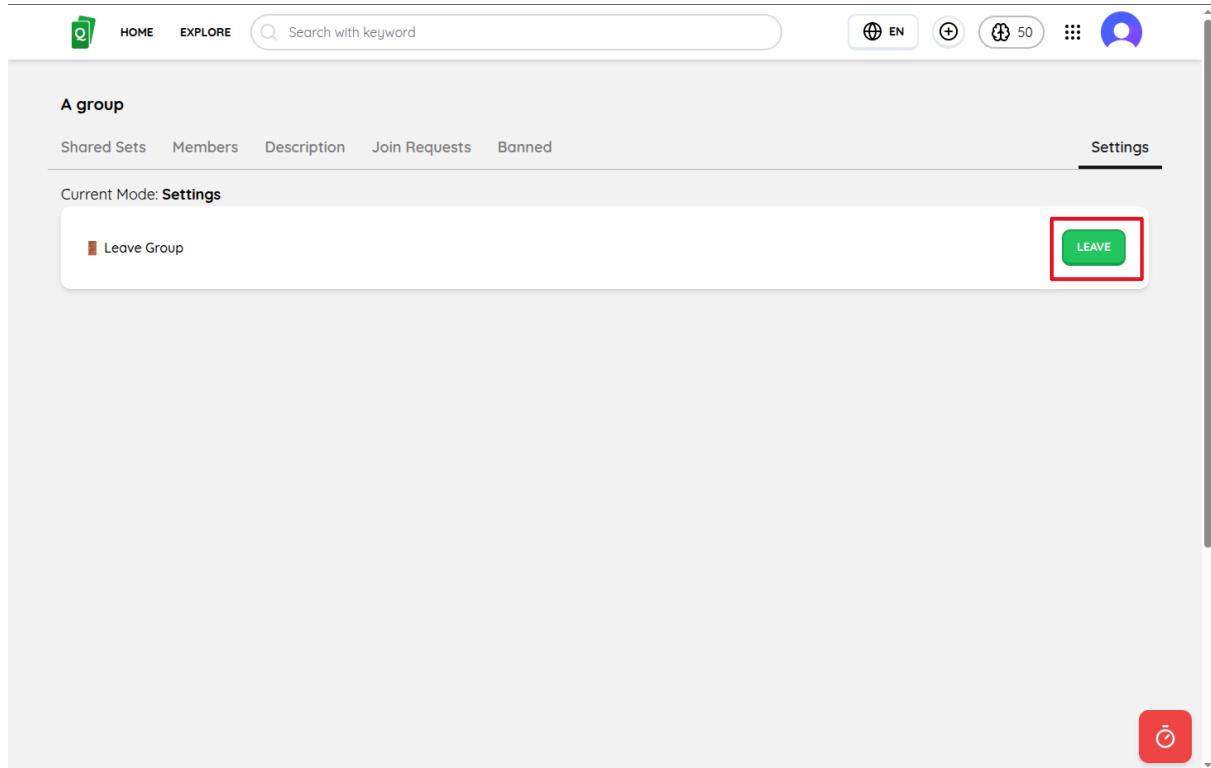


Figure 3.3.38 Leave groups

- Inside the group detail page, User clicks on the leave button to complete leaving a group.

3.3.42 Learner view goals

3.3.42.1 View created goals

The screenshot shows a user interface for a learning platform. At the top, there is a navigation bar with icons for HOME, EXPLORE, and a search bar. On the right side of the top bar, there are several buttons: EN (language), a plus sign, a circular icon with the number 50, a more options button (with a red box around it and the number 1 above it), and a user profile icon. A dropdown menu is open from the more options button, showing options: My Groups, My Sets, Goals (which has a red box around it and the number 2 above it), and Achievements.

Below the top bar, there is a goal card titled "Pick up where you left". The card includes a progress bar at 67% (Halfway done), a user icon for "thegaelord" (15 questions), a description about understanding basic physics principles, and categories: Classical Mechanics, Quantum Physics, Electromagnetism. It also shows a rating of 4.0 stars.

At the bottom of the screen, there is a quote by Benjamin Franklin: "Tell me and I forget. Teach me and I remember. Involve me and I learn." - Benjamin Franklin.

On the left side, there is a section titled "Data Structures and Algorithms" with two tabs: "Test Question Set #4" and "Test Question Set #8".

Figure 3.3.42.1.a: View goals - 1

- (1) User clicks on the more button
- (2) User clicks on the Goals button to be redirected to the goals page.

The screenshot shows the "Your goals" page. At the top, there is a header "Your goals" and a sub-header: "Using the goal-tracking feature, you can easily monitor your progress on personal projects and see how close you are to achieving your objectives." Below this, there are four status indicators: Not start (0), Finish (0), Doing (0), and Cancel (0). There is also a "Total" indicator (0).

Below the indicators, there is a "Status" filter set to "All" and a "CREATE" button. A single goal card is listed: "A goal" (Due date: May 8, 2025, Progress: 0%).

At the bottom right, there is a "1/1" indicator and a green download icon.

Figure 3.3.42.1.b: View goals - 2

- Inside the goals page, the user can see a list of their created goals.

3.3.42.2 View due goals

The screenshot shows a user interface for a digital learning platform. At the top, there is a navigation bar with icons for HOME, EXPLORE, a search bar labeled "Search with keyword", and various user account and system status indicators. Below the navigation bar, a section titled "Pick up where you left" displays three "Test Question Set"s:

- Test Question Set #1**: Not started (0%). Description: Explore fundamental concepts in mathematics, including algebra and geometry. Topics: Calculus, Linear Algebra, Statistics. Rating: ★★★★☆ 0.
- Test Question Set #3**: Not started (0%). Description: Learn about the fundamental concepts of biology, including cell structure and genetics. Topics: General Biology, Molecular Biology, Ecology. Rating: ★★★★☆ 0.
- Test Question Set #2**: Halfway done (67%). Description: Understand the basic principles of physics, from motion to energy. Topics: Classical Mechanics, Quantum Physics, Electromagnetism. Rating: ★★★★★ (4.0).

Below these sets, a reminder message says "Remember to do your goals!" followed by a box containing a single goal entry:

A goal	Doing
Due date: May 8, 2025	
Progress: 0%	

Figure 3.3.42.2 View goals

- User goes to the home page.
- If a created goal's deadline is due, a set of due goals will be displayed.

3.3.43 Learner view goal detail

The screenshot shows a user interface for managing goals. At the top, there are navigation links for 'HOME' and 'EXPLORE', a search bar, and various status indicators including 'EN', '50', and a profile icon. Below this, a section titled 'Your goals' displays a message about monitoring progress and achieving objectives. To the right, four status categories are shown with counts: 'Not start' (0), 'Finish' (0), 'Doing' (0), and 'Total' (0). A modal window titled 'A goal' is open, displaying the following details:

- Due date: Thu May 08 2025
- Current progress: 0% (indicated by a progress bar)
- Action buttons: 'CANCEL GOAL', 'FINISH' (highlighted in green), and 'DELETE'

The background shows a list of other goals with their respective due dates and progress percentages.

Figure 3.3.43 View goal detail

- User clicks on the goal card to view the goal detail.

3.3.44 Learner creates goals

The screenshot shows a user interface for managing goals. At the top, there is a navigation bar with icons for HOME, EXPLORE, and a search bar labeled "Search with keyword". On the right side of the top bar are buttons for EN, a plus sign, a profile icon, and a notification count of 50. Below the top bar, the main area is titled "Your goals". A sub-section titled "Using the goal-tracking feature, you can easily monitor your progress on personal projects and see how close you are to achieving your objectives." displays four status categories with counts: Not start (0), Finish (0), Doing (0), and Cancel (0). Below these stats is a large, empty rectangular box with a dashed border. In the top right corner of this box, there is a small number "1" followed by a blue "CREATE" button. This "CREATE" button is highlighted with a red rectangular box. At the bottom of the main area, there is a status filter dropdown set to "All" and a small "Y" icon. The bottom right corner of the main area has a red circular icon with a white letter "O".

Figure 3.3.44.a: Create goals - 1

- (1) User clicks on the create button.

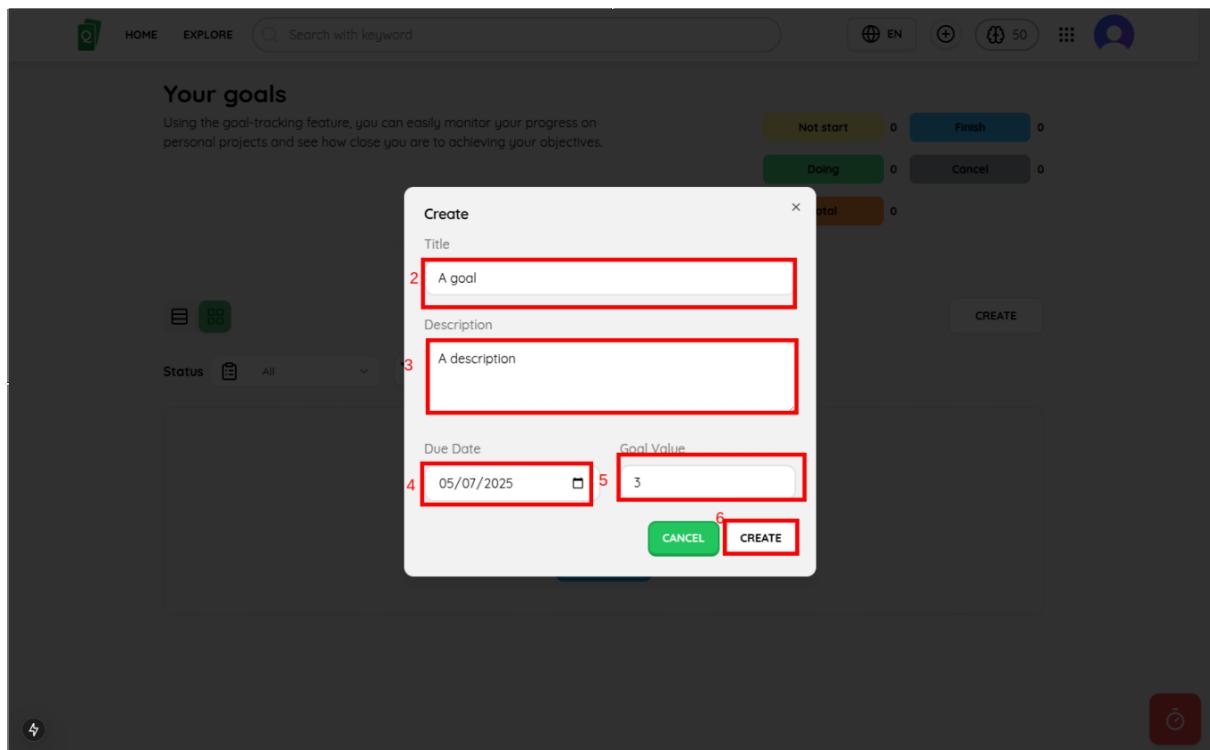


Figure 3.3.44.a: Create goals - 1

- (2) User inputs a title
- (3) User inputs a description
- (4) User chooses a due date
- (5) User inputs a goal value
- (6) User clicks the create button to finish creating a goal.

3.3.45 Learner update goal statuses

3.3.45.1 Update goal progress

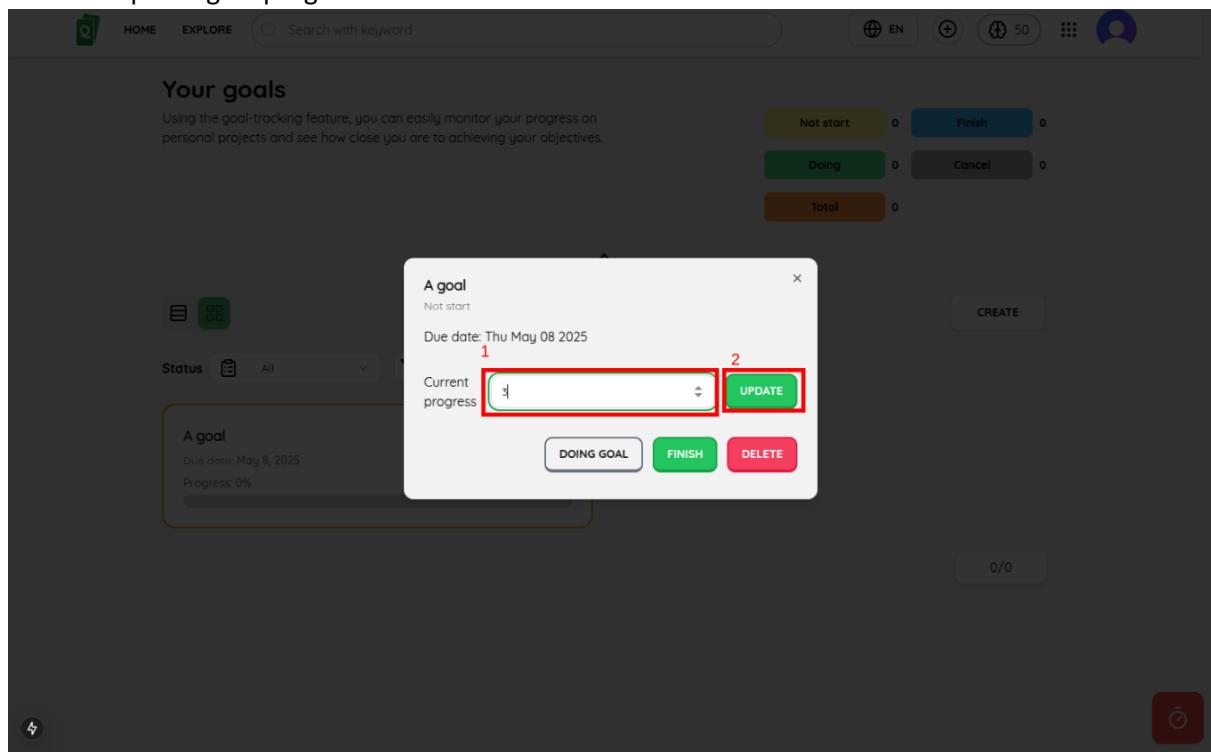


Figure 3.3.45.1 Update goal statuses

- (1) Inside the goal card, user inputs a progress value
- (2) User clicks the update button to complete updating goal progress

3.3.45.3 Cancel goal

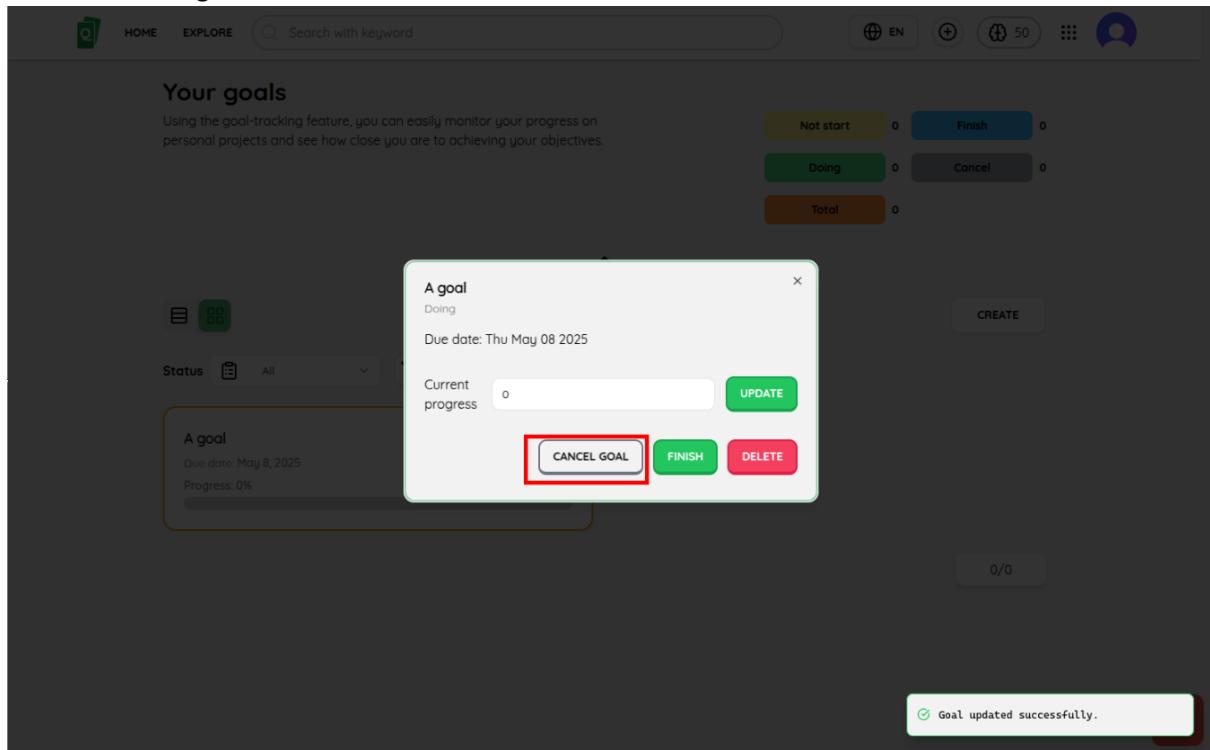


Figure 3.3.45.3 Update goal statuses

- This feature allows the user to mark a goal as cancelled, without completely deleting it.
- Inside goal detail popup, the user clicks on the cancel button to cancel goal progress.
- To cancel a goal, user has to set a goal as “doing” first

3.3.45.3 Finish goal progress

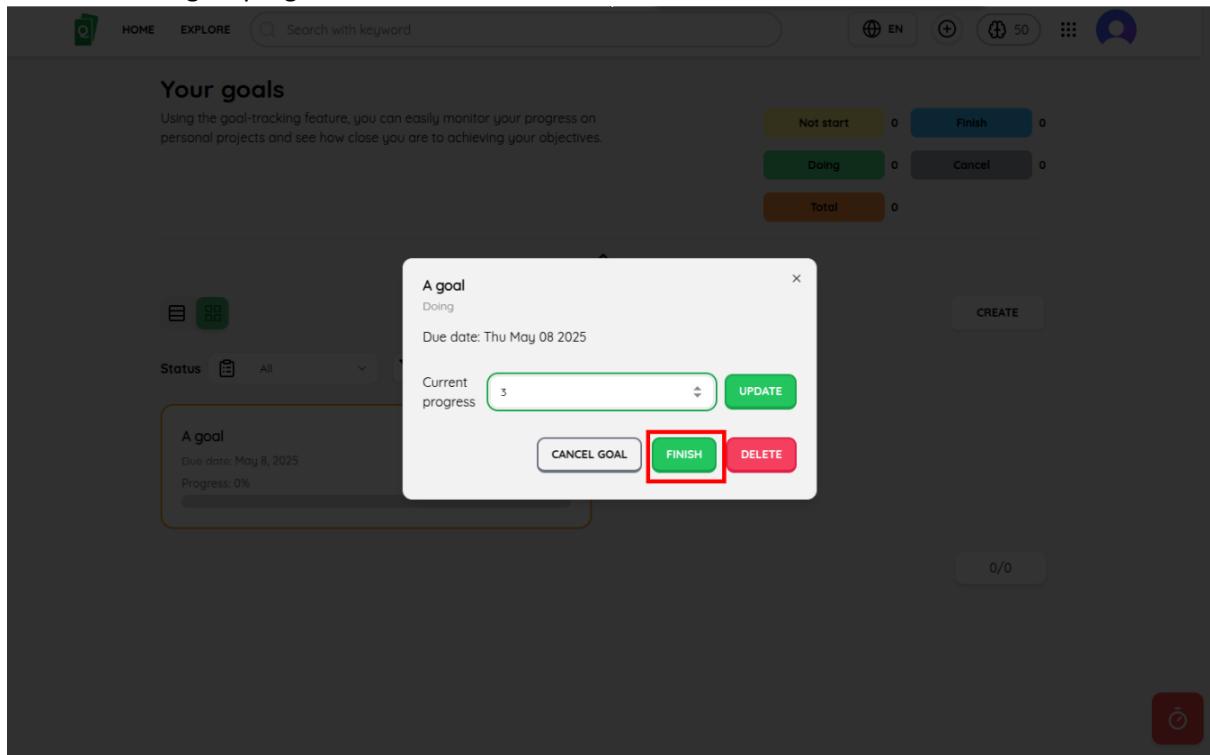


Figure 3.3.45.3 Update goal statuses

- This feature allows the user to complete a goal without reaching 100% progress.
- Inside the goal popup, the user clicks on the Finish button to finish goal progress.

3.3.46 Learner deletes goals

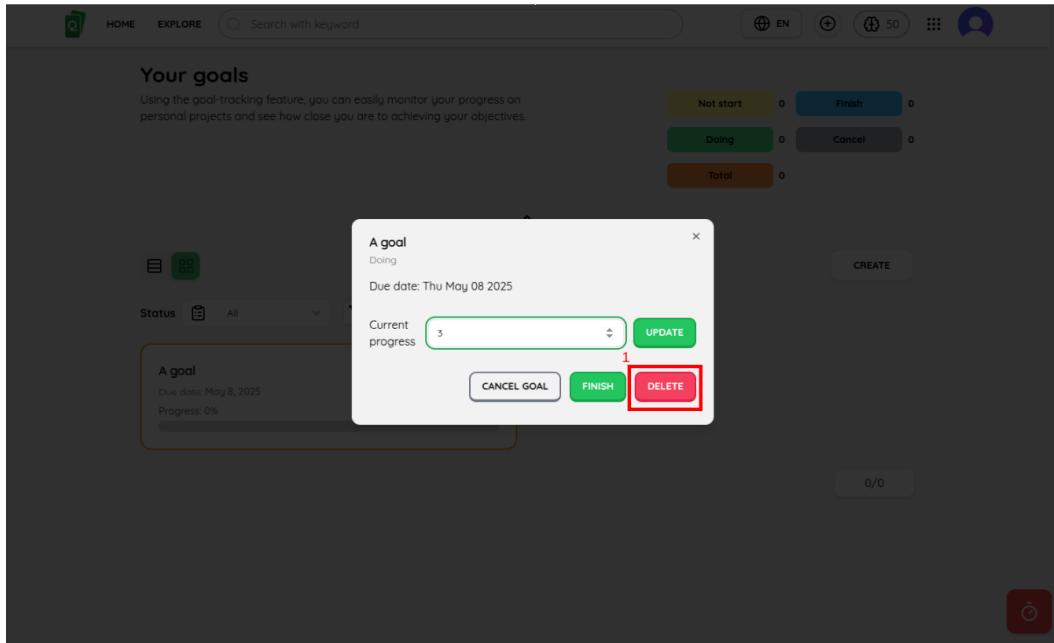


Figure 3.3.46.a Delete goals - 1

- (1) Inside the goal popup, the user clicks on the Delete button.

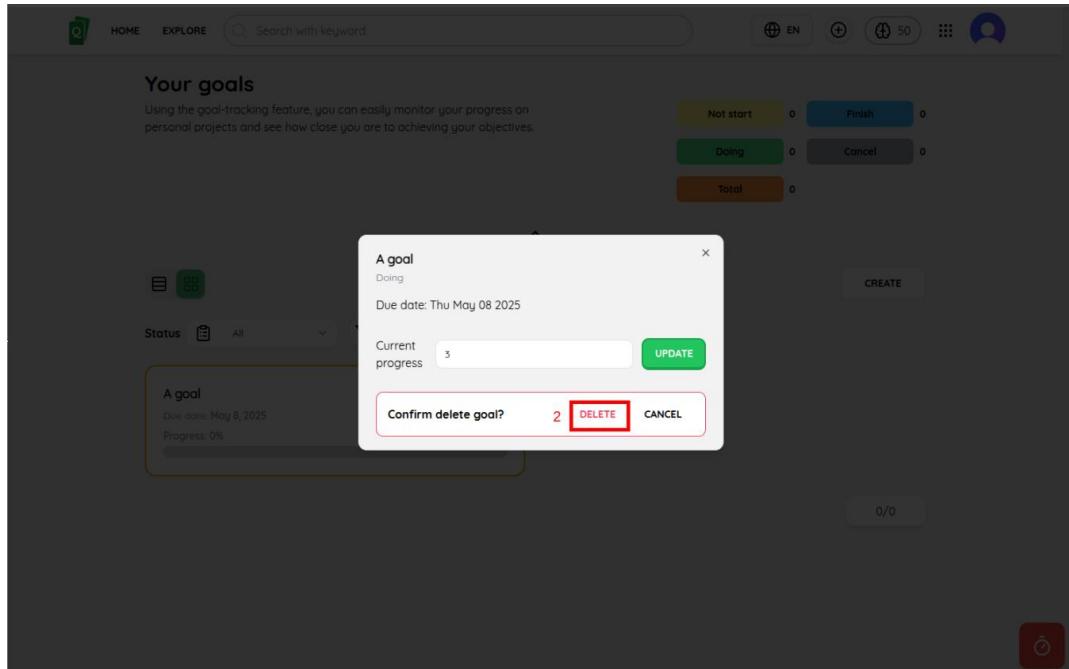


Figure 3.3.46.b Delete goals - 2

- (2) User clicks on the delete button to confirm delete a goal

3.3.47 Learner view settings

The screenshot shows a user interface for a learning platform. At the top, there is a navigation bar with icons for HOME, EXPLORE, and a search bar labeled "Search with keyword". On the right side of the bar are buttons for EN (language), a plus sign, a gear icon with the number 50, and a user profile icon with a red notification badge containing the number 1. A dropdown menu is open from the profile icon, showing options: Profile, Enable dark theme, Settings (which is highlighted with a red box and has a red number 2 next to it), and Sign out.

The main content area displays a "Test Question Set #2" card. The card includes the title "Test Question Set #2", a progress bar at 67% (labeled "Halfway done"), the author "thegelord", the count "15 questions", a description "Understand the basic principles of physics, from motion to energy.", and categories "Classical Mechanics", "Quantum Physics", and "Electromagnetism". Below the card is a rating of "★★★★★ (4.0)".

A reminder message "Remember to do your goals!" is displayed, followed by a goal card titled "A goal" with a due date of "May 8, 2025" and a progress of "0%".

Figure 3.3.47.a: View settings - 1

- (1) User clicks on the profile image.
- (2) User clicks on the settings button to be redirected to the settings page.

The screenshot shows the "Settings" page. On the left, there is a sidebar with links for Profile, System, and Account. The main content area is titled "Settings".

Profile Settings: Control and customize how your online presence is displayed and managed on this platform.

Fields include:

- Username: nta123
- Email: hnglabao11122003@gmail.com
- Full Name: (empty)
- Bio: Test bio #2

Buttons include:

- Show Full Name: On
- Show Email: On
- EDIT PROFILE button

System settings: Control and customize how the system is displayed on this platform.

Theme: LIGHT

Figure 3.3.47.b: View settings - 2

3.3.48 Change themes

3.3.48.1 Administrator change theme

The screenshot shows the Quizzzee administrator dashboard. On the left, a sidebar lists navigation items: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. Below these are Account management options: Tiếng Việt (highlighted with a red box and number 2), Dark theme (highlighted with a red box and number 2), and Sign out. A user account section shows a profile picture, the name 'nta123', and a dropdown arrow (highlighted with a red box and number 1). The main dashboard area has a 'Last Week' time filter. It displays four summary cards: 'Total learners 0' (blue icon), 'Total question sets 0' (purple icon), 'Total Groups 0' (orange icon), and 'Total Reports 0' (red icon). Below these are sections for 'Reports' (empty) and 'User accounts' (empty). A 'User Status' section is also present.

Figure 3.3.48.1 Change themes

- (1) User clicks on the button next to the account image.
- (2) User chooses the theme option inside the dropdown to complete changing a theme.

3.3.48.2 Learner change theme using header

The screenshot shows a user interface for a learning platform. At the top, there is a navigation bar with icons for HOME, EXPLORE, and a search bar labeled "Search with keyword". On the right side of the header, there are several buttons: EN (language), a plus sign, a circular icon with the number 50, a profile icon with a red notification badge (number 1), and a menu icon. A dropdown menu is open, showing options: Profile, Enable dark theme (which is highlighted with a red border and has a red number 2 next to it), Settings, and Sign out.

The main content area displays a "Test Question Set #2" titled "Pick up where you left". The set is 67% complete ("Halfway done"). It is created by "thegelord" and contains 15 questions. The description states: "Understand the basic principles of physics, from motion to energy." Below the description are three categories: Classical Mechanics, Quantum Physics, and Electromagnetism. The average rating is 4.0 stars.

Below the test set, a section titled "Remember to do your goals!" is shown. It contains a single goal: "A goal" due on May 8, 2025, with 0% progress indicated by a grey bar.

Figure 3.3.48.2 Change themes

- (1) User clicks on the profile image.
- (2) User clicks on the theme button to complete changing a theme.

3.3.48.3 Learner change theme inside the Settings page

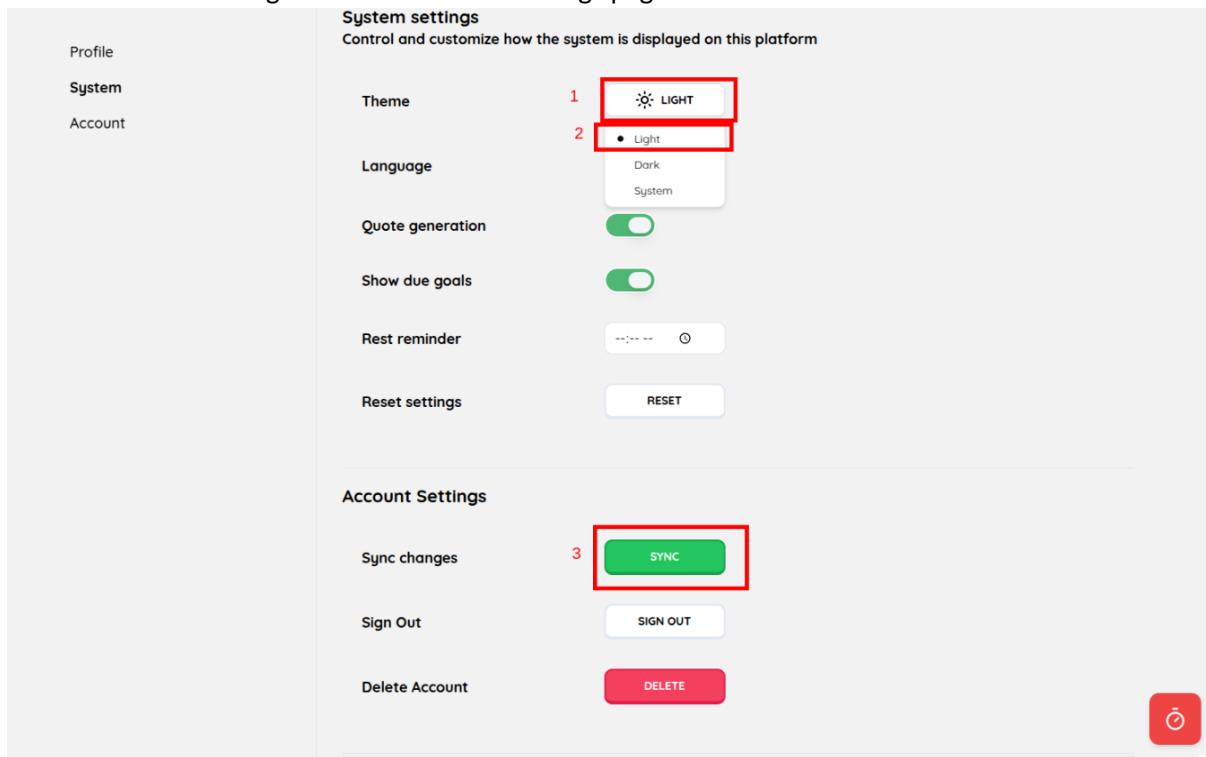


Figure 3.3.48.3 Change themes

- User scrolls to change the theme section.
- (1) User clicks on the change theme button
- (2) User clicks on an option.
- (3) User clicks on the save button to complete changing a theme.

3.3.49 Change languages

3.3.49.1 Administrator change language

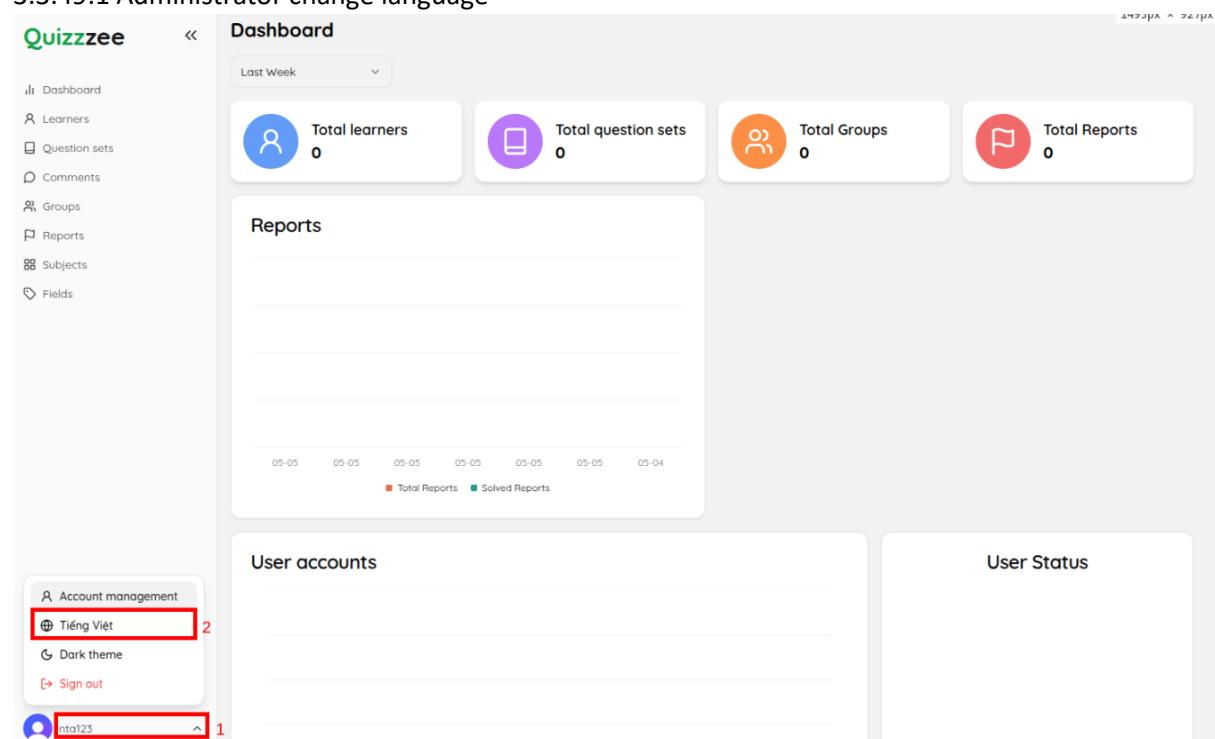


Figure 3.3.49.1 Change languages

- (1) User clicks on the button next to the account image.
- (2) User chooses the language option inside the dropdown to complete changing a language.

3.3.49.2 Learner and guest change language using header

The screenshot shows a user interface for a learning platform. At the top, there is a navigation bar with 'HOME' and 'EXPLORE' buttons, a search bar labeled 'Search with keyword', and a user profile icon. To the right of the search bar is a language selection area. Step 1 highlights the 'EN' button, which is currently selected and has a red border. Step 2 highlights the 'English' option in a dropdown menu, also with a red border. Below the header, a section titled 'Pick up where you left' contains a 'Test Question Set #2'. This section includes a progress bar indicating '67% Halfway done', a list of topics ('Classical Mechanics', 'Quantum Physics', 'Electromagnetism'), and a rating of '4.0'. A reminder 'Remember to do your goals!' is present, with a goal card showing 'A goal' due on 'May 8, 2025', 'Doing' status, and '0%' progress.

Figure 3.3.49.2 Change languages

- (1) User clicks on the language button.
- (2) User clicks on the language option to complete changing a language.

3.3.49.3 Learner change language inside the Settings page

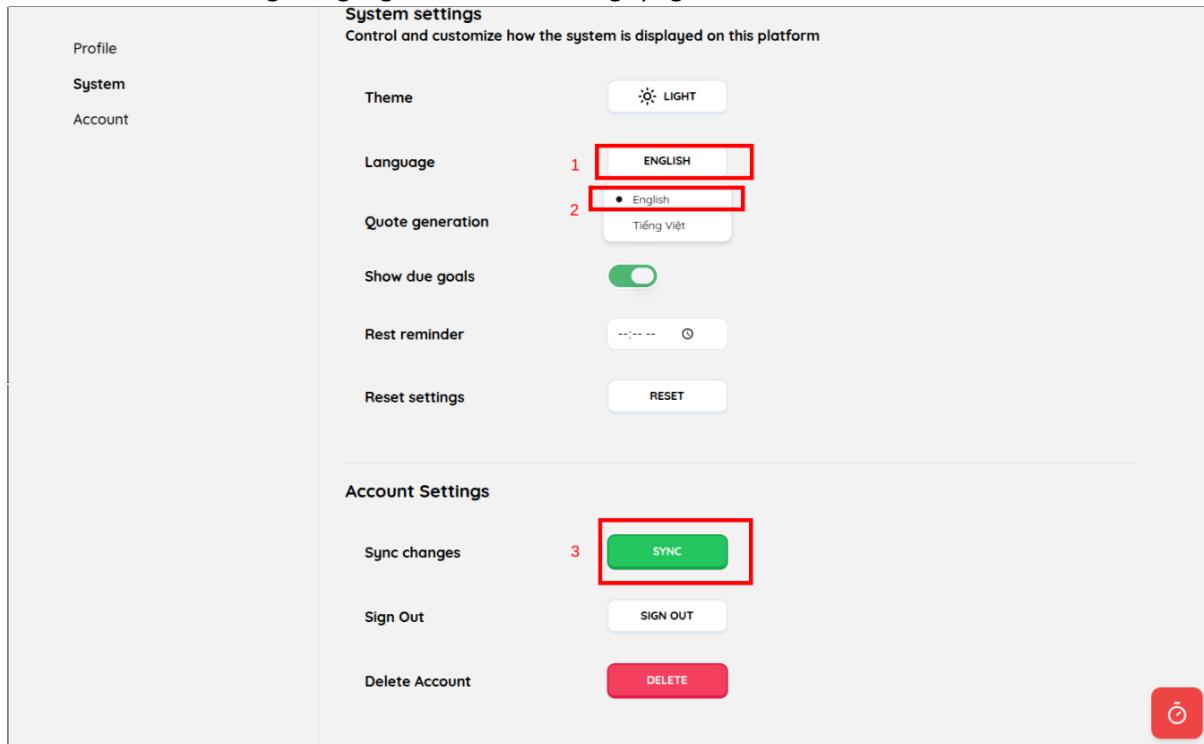


Figure 3.3.49.3 Change languages

- User scrolls to the change language page.
- (1) User clicks on the language button.
- (2) User clicks on the language option.
- (3) User clicks on the save button to complete changing a language.

3.3.50 Set rest reminder

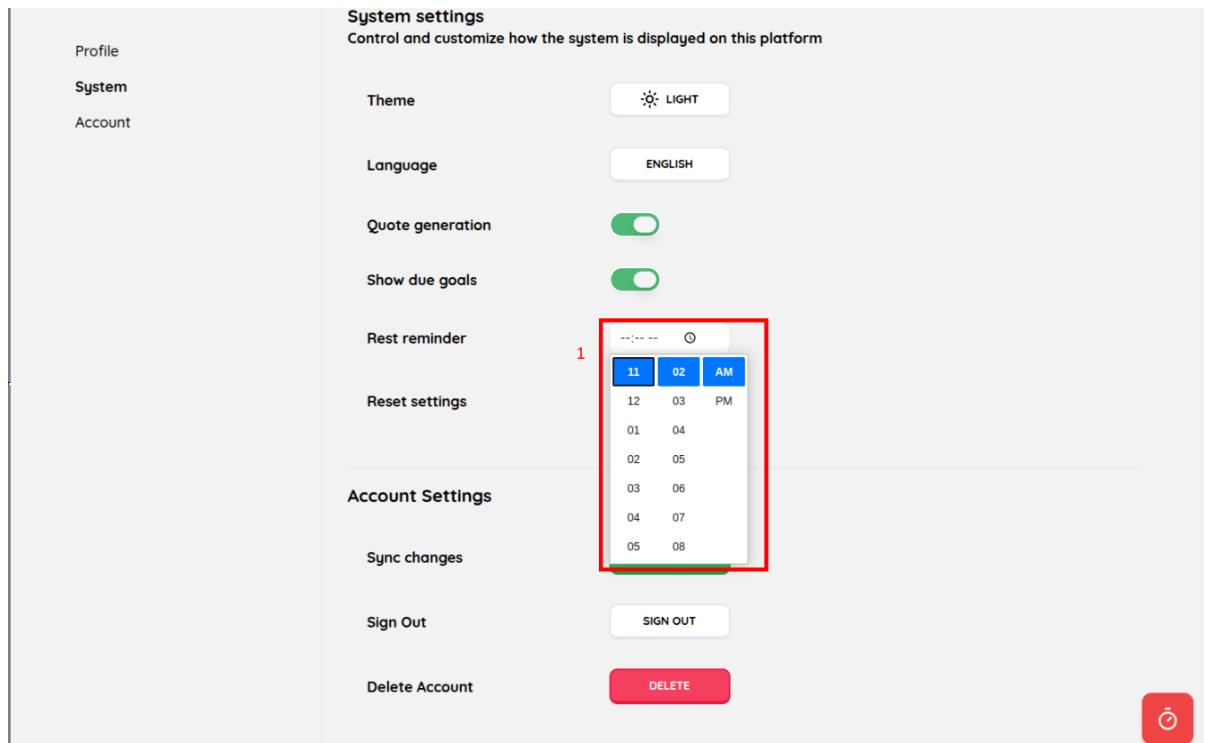


Figure 3.3.50.a. Set rest reminder - 1

- User scrolls to the reminder section.
- (1) User changes the time prompt.

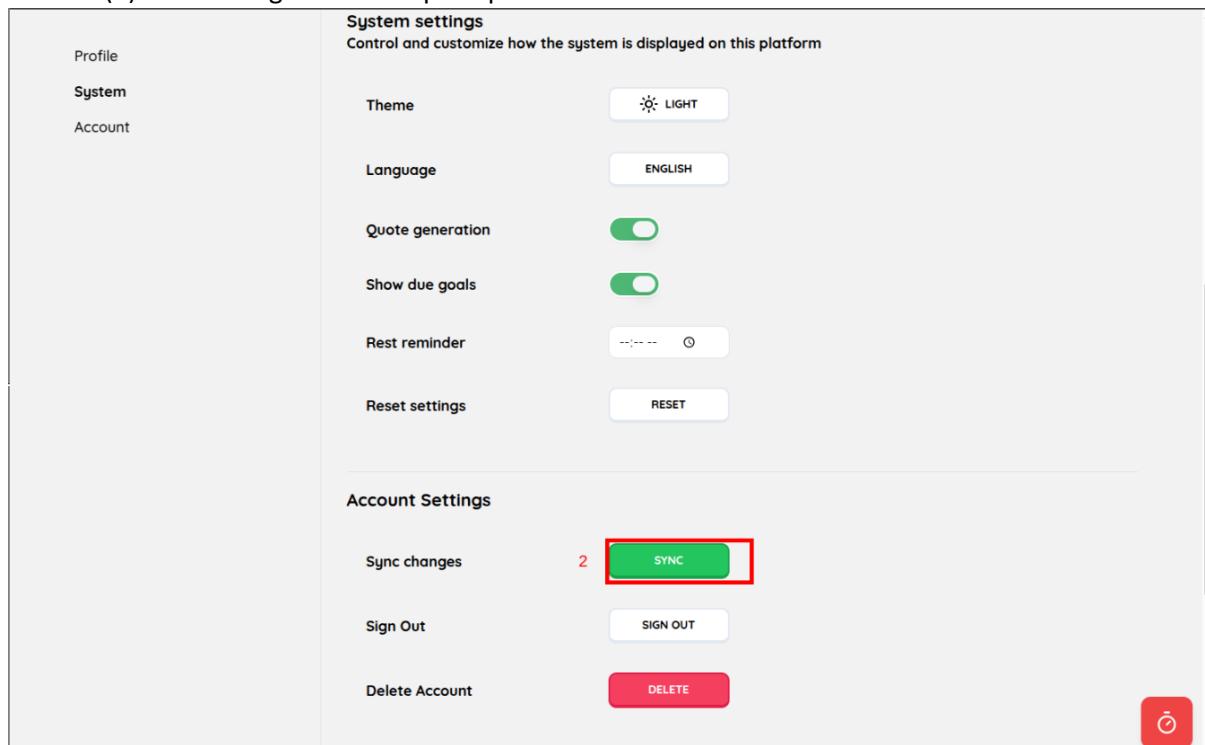
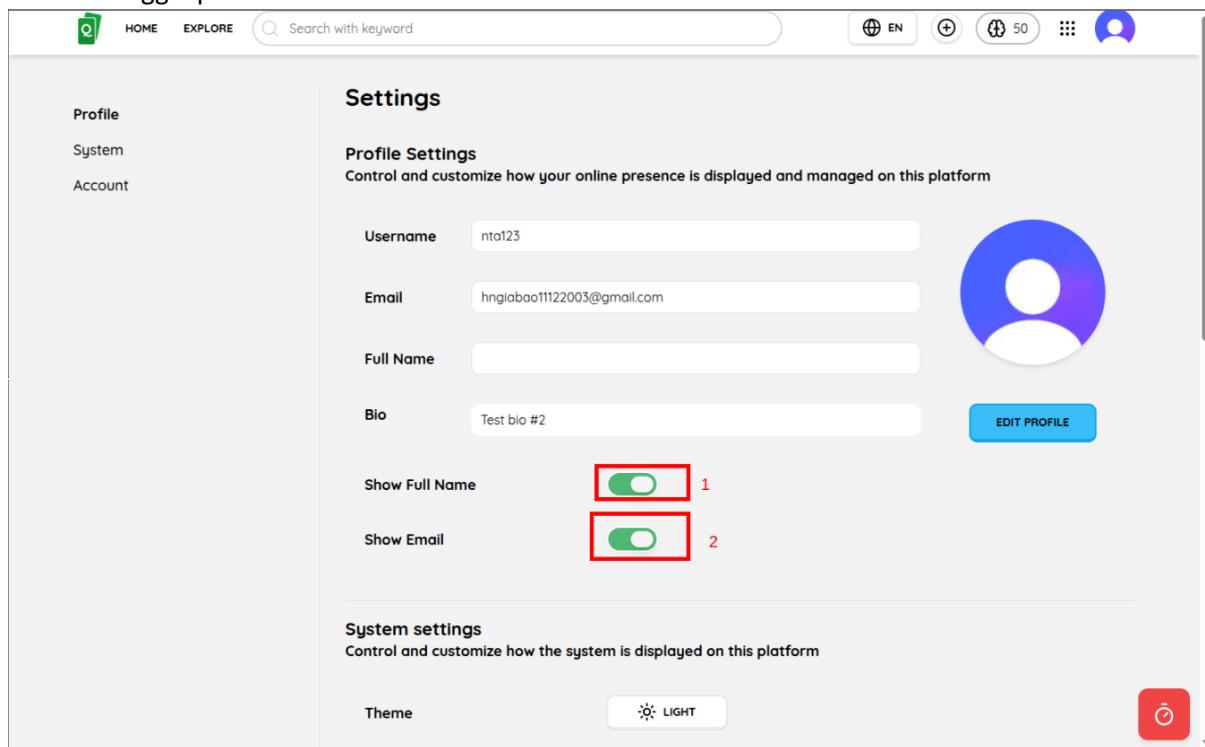


Figure 3.3.50.b. Set rest reminder - 2

- (2) User clicks on the save button to complete setting the rest reminder.
- A modal will pop up at the given time to remind the user to rest.

3.3.51 Learner toggle component

3.3.51.1 Toggle personal information



The screenshot shows a user profile settings page. At the top left, there are navigation links for 'HOME' and 'EXPLORE', a search bar with placeholder text 'Search with keyword', and several user-related icons including a profile picture, a gear, and a number '50'. The main content area has a sidebar on the left with 'Profile', 'System', and 'Account' options. The main panel is titled 'Settings' and contains the 'Profile Settings' section. This section is described as 'Control and customize how your online presence is displayed and managed on this platform'. It includes fields for 'Username' (nta123), 'Email' (hngiabao11122003@gmail.com), 'Full Name' (empty), and 'Bio' (Test bio #2). There are two toggle switches: 'Show Full Name' (which is turned on) and 'Show Email' (which is turned on). Both of these switches are highlighted with a red box and labeled '1' and '2' respectively. An 'EDIT PROFILE' button is located at the bottom right of this section. Below this is another section titled 'System settings' with the description 'Control and customize how the system is displayed on this platform'. It includes a 'Theme' switch set to 'LIGHT'.

Figure 3.3.51.1.a Toggle personal information - 1

- User scrolls to the toggle personal information section.
- (1) User clicks on the switch to hide full name. (optional)
- (2) User clicks on the switch to hide the email. (optional)

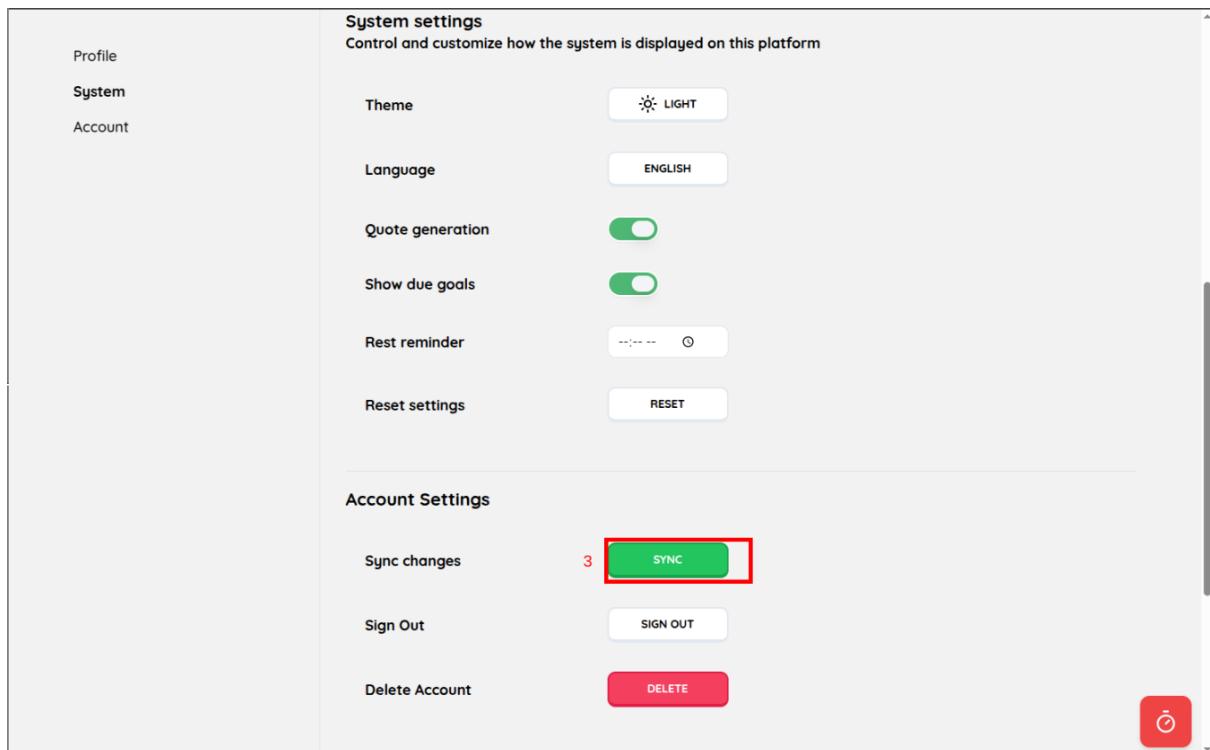


Figure 3.3.51.1.b Toggle personal information - 2

- (3) User clicks on the save button to complete setting the rest reminder.

3.3.51.2 Toggle system component

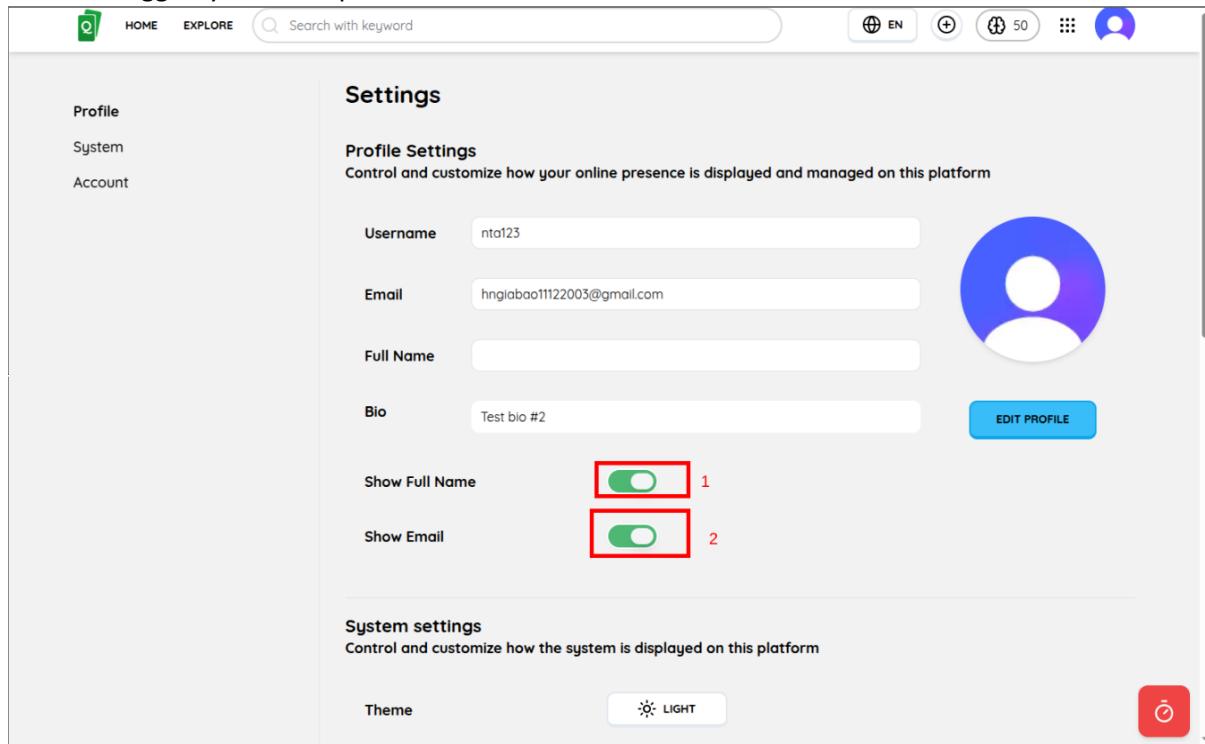


Figure 3.3.51.2 Toggle system component

- User scrolls to the toggle component section.
- (1) User clicks on the switch to hide quotes. (optional)
- (2) user clicks on the switch to hide due goals. (optional)

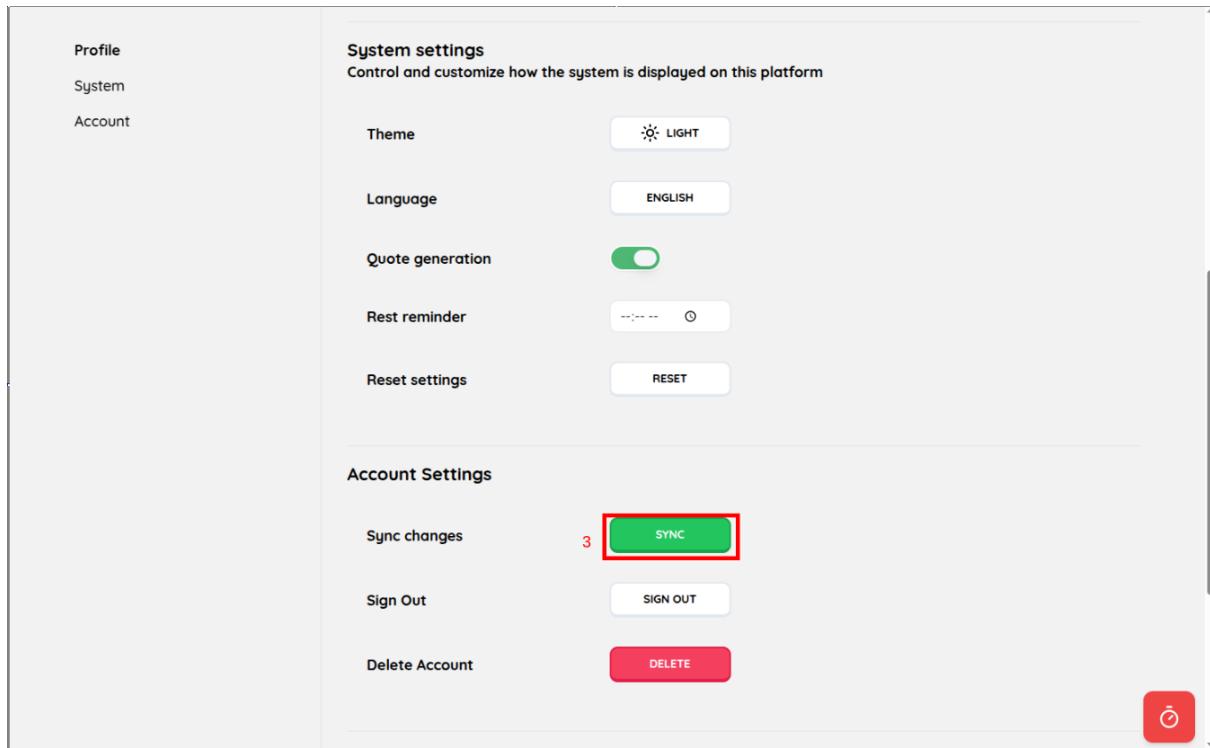


Figure 3.3.51.2.b Toggle system component - 2

- (3) User clicks on the save button to complete setting the rest reminder.

3.3.52 Learner reset settings

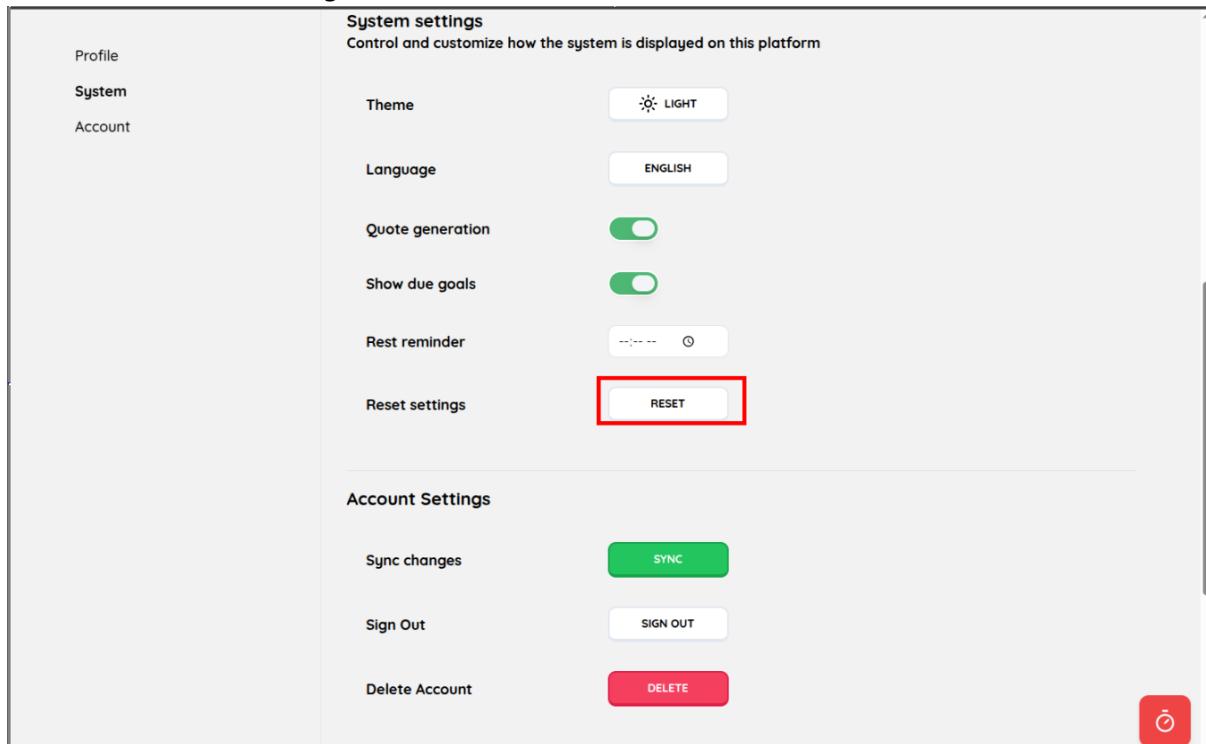


Figure 3.3.52 Reset settings

- User scrolls to the reset section.
- User clicks on the reset button to complete resetting settings.

3.3.53 View learners

3.3.53.1 Learner and Guest view learners from the search result page

The screenshot shows a web-based learning platform interface. At the top, there is a navigation bar with icons for HOME, EXPLORE, and a search bar containing the placeholder "Search with keyword". A red box highlights the search bar. To the right of the search bar are various global settings and user profile icons. Below the navigation bar, a sidebar titled "Pick up where you left" displays a list of "Test Question Set #2" by user "thegaelord". The set contains 15 questions and is described as "Halfway done". It includes categories like Classical Mechanics, Quantum Physics, and Electromagnetism, and has a rating of 4.0 stars. In the center of the page, there is a quote by Mahatma Gandhi: "Live as if you were to die tomorrow. Learn as if you were to live forever." At the bottom, there are links for "Data Structures and Algorithms", "Test Question Set #4", and "Test Question Set #8", along with a red circular icon.

Figure 3.3.53.1.a: View learner - 1

- (1) User clicks on the search bar.

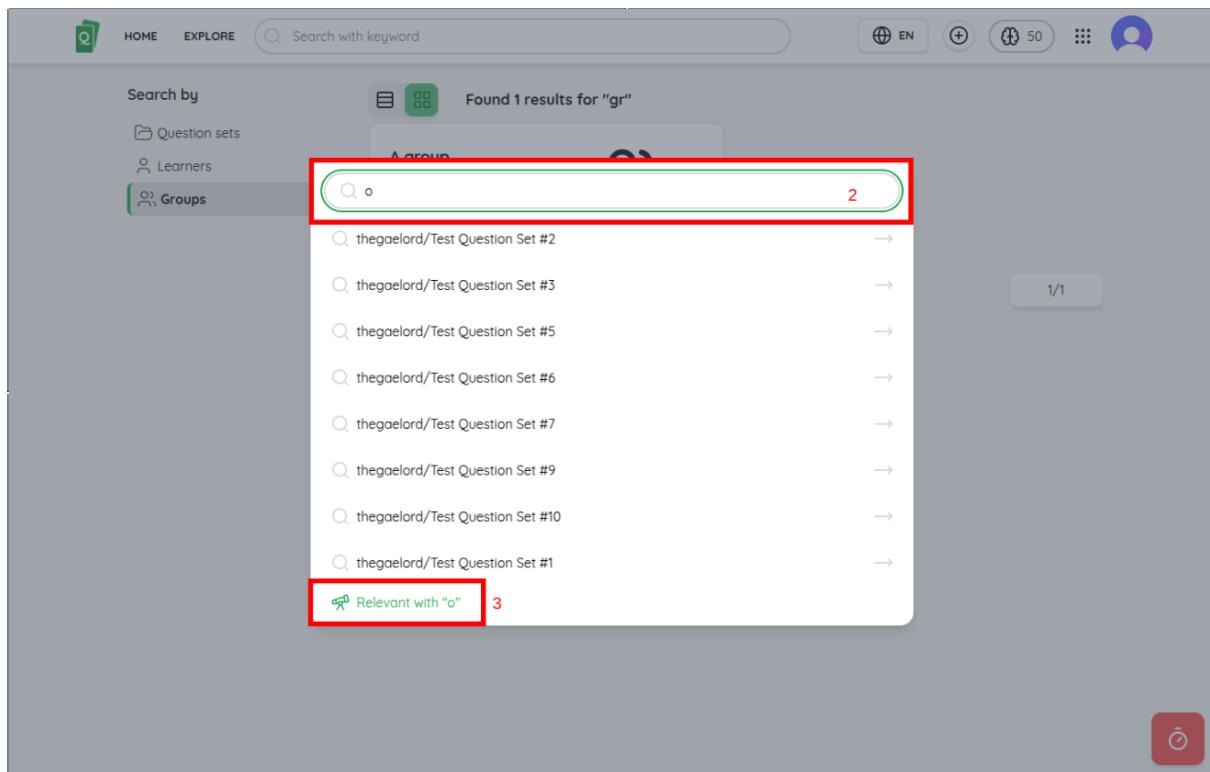


Figure 3.3.53.1.b: View learners - 2

- (2) User inputs keyword inside search bar.
- (3) User presses Enter or clicks the relevant link to be redirected to the search results page.

The screenshot shows a user interface for a digital platform. At the top, there are navigation links for 'HOME' and 'EXPLORE', a search bar with the placeholder 'Search with keyword', and various user account icons. Below the search bar, a section titled 'Found 8 results for "U"' displays a list of users. The first result is a profile for 'Huynh Nguyen Gia Bao (K17 CT)' with the handle '@thegoelord'. The profile picture is a penguin. To the left of the search results, a sidebar titled 'Search by' lists several categories: 'Question sets', 'Learners' (which is highlighted with a red box), 'Groups', and 'interested' (with sub-options like Biology, Mathematics, Physics, Chemistry, History, Geography, Literature, Computer Science, Economics, and Language). On the right side of the search results, there is a small box labeled 'First Name' with the value '@user3' and '1 follower'. The bottom right corner of the interface has a red circular button with a white icon.

Figure 3.3.53.1.c: View learners - 3

- (4) User clicks on the learner filter to view learners.

3.3.53.2 Administrators views learners from the learners page

The screenshot shows the Quizzzee application interface. On the left, a sidebar menu includes options like Dashboard, Learners (which is highlighted with a red box), Question sets, Comments, Groups, Reports, Subjects, and Fields. The main content area is titled "Learners" and displays a table of three learner records. The table columns are: USERNAME, Email, Full Name, Active Status, NEURON, Created At, and Updated At. The data in the table is as follows:

USERNAME	Email	Full Name	Active Status	NEURON	Created At	Updated At
thegaelord	baohngce171025@fpt.edu.vn	Huynh Nguyen Gia Bao (KT7 CT)	Active	0	04/30/2025	04/30/2025
nta123	hngiabao11122003@gmail.com		Active	50	04/30/2025	05/04/2025
user3	user3@example.com	First Name	Active	0	04/30/2025	04/30/2025

At the bottom right of the table, there are "PREVIOUS" and "NEXT" buttons. A user profile icon for "nta123" is visible at the bottom left.

Figure 3.3.53.2 View learners

- User clicks on the learners button to go to the learners page.

3.3.54 View learner profile

3.3.54.1 Learners and guests view learner profile from search result page

The screenshot shows a search results page with a search bar at the top containing the keyword "u". A red box highlights a learner profile card for "Huynh Nguyen Gia Bao (K17 CT) @thegaelord". The card features a penguin icon, the user's name, their CT status, and their GitHub handle. It also shows they have 1 follower. To the right of the card, there is a sidebar with a "First Name" section showing "@user3" and "1 followers". The left sidebar lists categories under "interested" such as Biology, Mathematics, Physics, Chemistry, History, Geography, Literature, Computer Science, Economics, and Language.

Figure 3.3.54.1 View learner profile - a

- Inside the search result page, the user clicks on the learner card to be redirected to the learner profile page.

The screenshot shows a detailed learner profile for "Huynh Nguyen Gia Bao (K17 CT) @thegaelord". The profile includes a large penguin icon, the user's name, their CT status, and a "FOLLOW" button. Below the profile, there are sections for "Test bio #1" (0 followers, 0 followees), "Interested Fields" (Calculus, Linear Algebra, Statistics), and "Achievements". To the right, there are six "Test Question Set" cards, each with a title, author, number of questions, a brief description, and a list of topics. Each set has a 5-star rating below it.

Figure 3.3.54.2 View learner profile - b

3.3.54.2 Learners and guests view learner profile from question set detail page

The screenshot shows a user interface for a question set detail page. At the top, there is a navigation bar with icons for HOME, EXPLORE, and a search bar. On the right side of the header are language settings (EN), a notification icon (50), and a user profile icon.

The main content area is titled "Test Question Set #1". It includes a brief description: "About this question set: Explore fundamental concepts in mathematics, including algebra and geometry.", the last update date (2025-05-05), a rating of 4.0 stars, and tags for Calculus, Linear Algebra, and Statistics.

A user profile card for "thegaelord" is displayed, featuring a small profile picture, the username "thegaelord", and a "FOLLOW" button. This card is highlighted with a red rectangular box.

Below the profile card is a "Learning Progress" section with a large orange circle representing the progress status. To the right of the circle are four buttons: FLASH CARD, PRACTICE, QUIZ, and PLAY. A legend below the circle indicates colors for Not Learned (red), Learning (green), Almost done (blue), and Mastered (yellow).

At the bottom left, it says "21 terms". On the right, there are "BASIC SEARCH" and a magnifying glass icon.

Figure 3.3.54.2.a View learner profile - 1

- Inside the question set detail page, the user clicks on the leaner username to be redirected to the learner profile page.

The screenshot shows a learner profile page for "Huynh Nguyen Gia Bao (K17 CT) @thegaelord". The profile picture is a Tux icon. Below it, the name "Huynh Nguyen Gia Bao (K17 CT)" and the handle "@thegaelord" are displayed, along with a "FOLLOW" button.

On the left, there is a sidebar with sections for "Interested Fields" (Calculus, Linear Algebra, Statistics) and "Achievements".

The main content area displays six question sets in cards:

- Test Question Set #1**: Created by @thegaelord, 21 questions. Description: Explore fundamental concepts in mathematics, including algebra and geometry. Tags: Calculus, Linear Algebra, Statistics. Rating: 4.0 stars.
- Test Question Set #2**: Created by @thegaelord, 15 questions. Description: Understand the basic principles of physics, from motion to energy. Tags: Classical Mechanics, Quantum Physics, Electromagnetism. Rating: 4.0 stars.
- Test Question Set #3**: Created by @thegaelord, 6 questions. Description: Learn about the fundamental concepts of biology, including cell structure and genetics. Tags: Molecular Biology, Ecology, General Biology. Rating: 4.0 stars.
- Test Question Set #4**: Created by @thegaelord, 0 questions. Description: Dive into the basics of computer science, covering algorithms and data structures. Tags: Data Structures and Algorithms, Artificial Intelligence, Database Systems. Rating: 4.0 stars.
- Test Question Set #5**: (partially visible)
- Test Question Set #6**: (partially visible)

Figure 3.3.54.2.b View learner profile - 2

3.3.54.3 Administrator views learners from the learners page

USERNAME	Email	Full Name	Active Status	NEURON	Created At	Updated At
thegaelord	baohngce171023@fpt.edu.vn	Huynh Nguyen Gia Bao (K17 CT)	Active	0	04/30/2025	04/30/2025
nta123	hngiabao11122003@gmail.com		Active	50	04/30/2025	05/04/2025
user3	user3@example.com	First Name	Active	0	04/30/2025	04/30/2025

Figure 3.3.54.3.a View learner profile - 1

- User clicks on the learner table row to be redirected to the learner profile page.

Huynh Nguyen Gia Bao (K17 CT)

thegaelord
baohngce171023@fpt.edu.vn

Block User active

Personal Information

Username	thegaelord
Email	baohngce171023@fpt.edu.vn
Language	en
Theme	dark

Account Status

Status	online
Created	Apr 30, 2025
Last Updated	Apr 30, 2025

Bio

Test bio #1

Figure 3.3.54.3.b View learner profile - b

3.3.55 Learner follow learners

The screenshot shows a user profile page for 'Huynh Nguyen Gia Bao (K17 CT)'. The profile picture is a cartoon penguin wearing a red cap with 'deban' on it. Below the profile picture, the user's name and handle (@thegaelord) are displayed. A large red box highlights the 'FOLLOW' button, which is currently white with black text. To the right of the profile picture, there is a summary section with 'Test Question Set #1' and 'Test Question Set #2'. Further down, there are sections for 'Achievements', 'Interested Fields', and 'Test Question Set #3' through '#6'. Each section contains a brief description, the user's handle (@thegaelord), the number of questions, and a list of topics or fields.

HOME EXPLORE Search with keyword EN + 50 Groups

Huynh Nguyen Gia Bao (K17 CT)
@thegaelord FOLLOW

Test bio #1
0 followers 0 followees

Interested Fields
Calculus Linear Algebra Statistics

Achievements

Test Question Set #1
@thegaelord 21 questions
Explore fundamental concepts in mathematics, including algebra and geometry.
Calculus Linear Algebra Statistics

Test Question Set #2
@thegaelord 15 questions
Understand the basic principles of physics, from motion to energy.
Classical Mechanics Quantum Physics Electromagnetism

Test Question Set #3
@thegaelord 6 questions
Learn about the fundamental concepts of biology, including cell structure and genetics.
Molecular Biology Ecology General Biology

Test Question Set #4
@thegaelord 0 questions
Dive into the basics of computer science, covering algorithms and data structures.
Data Structures and Algorithms Artificial Intelligence Database Systems

Test Question Set #5

Test Question Set #6

Figure 3.3.55. Follow learners

- Inside the learner profile page, the user clicks on the follow button to complete following another learner.
- Once follow, the follow button will change to “Following”

3.3.56 Learner unfollow other learners

The screenshot shows a learner profile page. At the top, there is a navigation bar with icons for HOME, EXPLORE, a search bar, and user settings. Below the navigation bar is the learner's profile section, which includes a penguin icon, the name 'Huynh Nguyen Gia Bao (K17 CT)', the handle '@thegaelord', and a 'FOLLOWING' button with a red border. Below this are sections for 'Test bio #1', 'Achievements', and 'Interested Fields'. To the right of the profile are several 'Test Question Set' cards, each with a title, author, number of questions, a brief description, and categories. The first card, 'Test Question Set #1', has a 'Calculus' category highlighted in pink.

Figure 3.3.56. Unfollow learners

- Inside the learner profile page, the user clicks on the following button to complete unfollowing another learner.
- Once unfollow, the follow button will change to “Follow”

3.3.57 View follow detail

3.3.57.1 Learners or guests view learner's followers detail

The screenshot shows a user profile for 'Huynh Nguyen Gia Bao (K17 CT)'. On the left, there is a circular profile picture of a penguin. Below it, the user's name and handle (@thegaelord) are displayed. A 'FOLLOWING' button is shown. In the center, there is a section titled 'Test bio #1' which includes a follower count of '1 follower' (highlighted with a red box). To the right, there are sections for 'Interested Fields' (Calculus, Linear Algebra, Statistics) and 'Achievements' (Test Question Set #1 through #6). The background features a light gray grid pattern.

Figure 3.3.57.1 View followers - 1

- Inside learner profile page, user clicks on the followers number to view followers detail

This screenshot is identical to Figure 3.3.57.1, but a modal window has been opened over the '1 follower' link. The modal title is 'Followers of thegaelord' and contains a single entry: 'nta123'. The rest of the page content is dimmed, indicating it is not active.

Figure 3.3.57.1.b View followers - 2

3.3.57.2 Administrator view learner's follow detail

USERNAME ↑↓	Email	Full Name	Active Status	NEURON ↑↓	Created At	Updated At
thegaelord	baohngce171023@fpt.edu.vn	Huynh Nguyen Gia Bao (K17 CT)	Active	0	04/30/2025	04/30/2025
nta123	hngiabao11122003@gmail.com		Active	50	04/30/2025	05/04/2025
user3	user3@example.com	First Name	Active	0	04/30/2025	04/30/2025

Figure 3.3.57.2 View follow detail

- (1) User clicks the learner table row inside the learner page.
- (2) User goes to the follow detail section to view a learner's follow detail.

3.3.58 Administrator view reports

The screenshot shows the Quizzzee application interface. On the left, there is a sidebar with the following navigation options: Dashboard, Learners, Question sets, Comments, Groups, Reports (which is highlighted with a red box), Subjects, and Fields. The main content area is titled "Reports" and displays a table with one row. The table columns are: Reported By, TYPE, Title, Description, Resolved, and Created At. The single row in the table shows: nta123, questionSet, An appropriate question set, There are some invalid information inside this set, No, and 5/5/2025, 3:57:20 PM. There is also a "Filter username..." input field above the table. At the bottom right of the table area, there are "PREVIOUS" and "NEXT" buttons.

Reported By	TYPE	Title	Description	Resolved	Created At
nta123	questionSet	An appropriate question set	There are some invalid information inside this set	No	5/5/2025, 3:57:20 PM

Figure 3.5.58 View reports

- User clicks on the reports button to be redirected to the reports page.
- Inside the reports page, the user can view a table of reports.

3.3.59 Administrator view report detail

The screenshot shows the 'Reports' section of the Quizzzee application. On the left is a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. The main area is titled 'Reports' and shows a table with one row. The table columns are: Reported By, TYPE, Title, Description, Resolved, and Created At. The single row contains: nta123, questionSet, An appropriate question set, There are some invalid information inside this set, No, and 5/5/2025, 3:57:20 PM. A red box highlights the entire row.

Figure 3.3.59.a: View report detail - 1

- User clicks on the report card to view the details of a report.

The screenshot shows the detailed view of a report for a question set. The top part is titled 'Report questionSet' and contains the following information: 'An appropriate question set' (Reported by: nta123), 'There are some invalid information inside this set' (Status: Pending), and a 'Mark as Resolved' button. Below this is a section titled 'Question Set Information' for 'Test Question Set #1'. It includes a description ('Explore fundamental concepts in mathematics, including algebra and geometry.'), an 'Author' field ('thegaelord'), a 'Rating' field, and a 'Subjects' field ('Calculus / Giải tích', 'Linear Algebra / Đại số tuyến tính', 'Statistics / Thống kê'). The bottom section is titled 'Questions (21)' and shows a sample question: 'What is the value of π (pi) to two decimal places?' (Type: multiple_choice). The correct answer is listed as '3.14'.

Figure 3.3.59.b: View report detail - 2

3.3.60 Learner creates reports

3.3.60.1 Report question sets

The screenshot shows a question set detail page titled "Test Question Set #3". The page includes a description, last update date (2025-05-05), rating (4.0 stars), and tags (General Biology, Molecular Biology, Ecology). A user profile "thegaelord" is shown with a "FOLLOW" button. On the right, there are several icons: a download arrow, a link, a report icon (highlighted with a red box and labeled "1"), and a heart. Below these are sections for "Learning Progress" (FLASH CARD, PRACTICE, QUIZ) and a rating scale from 1 to 5 stars.

Figure 3.3.60.1.a Create reports - 1

- (1) Inside the question set detail page, the user clicks on the report button.

The screenshot shows a "Report Question Set" modal window. It contains two input fields: "2 Inappropriate question set" and "3 There are some misleading information inside this set". At the bottom are "CANCEL" and "SUBMIT" buttons. The number "4" is located at the bottom right of the modal. The background shows the same question set detail page as Figure 1, with the report button now highlighted.

Figure 3.3.60.1.b Create reports - 2

- (2) User input title
- (3) User input description.
- (4) User clicks the report button to complete creating a report.

3.3.60.2 Report comments

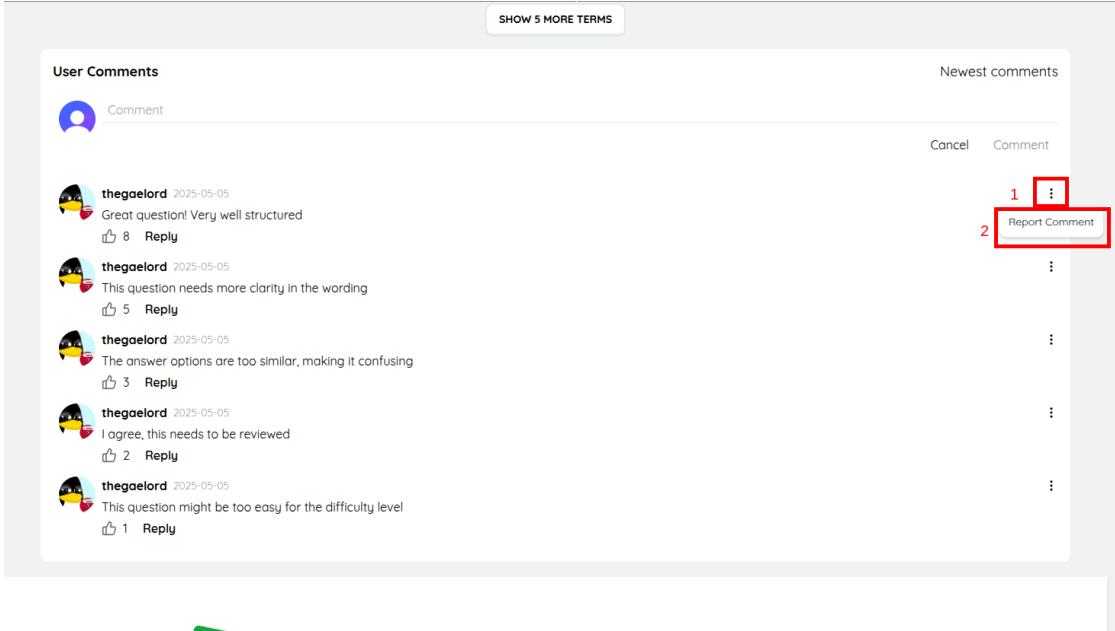


Figure 3.3.60.2.a: Create reports - 1

- Inside the question set detail page, the user scrolls to a comment.
- (1) User clicks on the dropdown button inside the comment card.
- (2) User clicks on the report button.

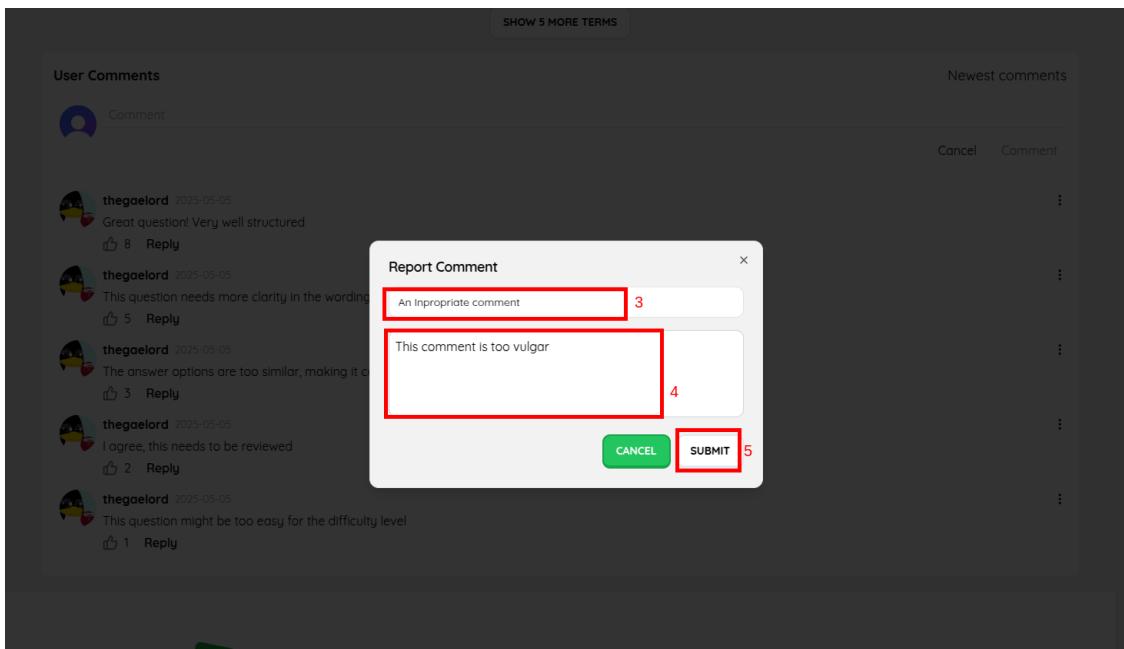


Figure 3.3.60.2.b: Create reports - 2

- (3) User input title
- (4) User input description.
- (5) User clicks the report button to complete creating a report.

3.3.61 Administrator update report statuses

The screenshot shows a user interface for managing reports. On the left is a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects (which is selected and highlighted in grey), and Fields. The main content area has a header "Report questionSet" and a sub-header "An appropriate question set". It states "Reported by: nta123" and "There are some invalid information inside this set". A red box highlights the "Pending" status and the "Mark as Resolved" button, which is step 2. Below this, it shows "Created at: 5/5/2025, 3:57:20 PM" and "Updated at: 5/5/2025, 3:57:20 PM". The "Question Set Information" section includes a "Test Question Set #1" title, a "Ban" button (step 1) with a red box around it, author information ("Author: thegaelord"), rating, subjects (Calculus / Giải tích, Linear Algebra / Đại số tuyến tính, Statistics / Thống kê), and a list of 21 questions. One question is visible: "What is the value of π (pi) to two decimal places?".

Figure 3.3.61 Update report statuses

- Inside the report detail page, the user can choose to ban or skip the report.
- (1) The user can choose to ban the reported subject by clicking the ban button.
- (2) User clicks on the resolve button to mark a report as resolved

3.3.62 Administrator update learner visibilities

The screenshot shows a user profile for 'Huynh Nguyen Gia Bao (K17 CT)' on the Quizzzee platform. The profile includes the following details:

Personal Information		Account Status	
Username	thegaelord	Status	online
Email	baohngce171023@fpt.edu.vn	Created	Apr 30, 2025
Language	en	Last Updated	Apr 30, 2025
Theme	dark		

Bio: Test bio #1

A red button labeled 'Block User' is positioned next to a green button labeled 'active'.

Figure 3.3.62. Update learner visibilities

- User clicks on the ban button inside a learner table row to complete banning a learner.
- By clicking the button again will unblock the learner.

3.3.63 Administrator update question set visibilities

The screenshot shows the Quizzzee application interface. On the left is a sidebar with navigation links: Dashboard, Learners, Question sets (highlighted), Comments, Groups, Reports, Subjects, and Fields. The main content area is titled "Test Question Set #1" and describes it as "Explore fundamental concepts in mathematics, including algebra and geometry." It shows the "Author" as "thegaelord" and a "Rating" section. A red box highlights the "Ban" button in the top right corner of the main content area. Below this, there's a "Subjects" section with categories: Calculus / Giải tích, Linear Algebra / Đại số tuyến tính, and Statistics / Thống kê. The "Questions (21)" section lists two questions:

- What is the value of π (pi) to two decimal places?
Type: multiple_choice
3.14 (selected)
3.15
3.13
3.16
- What is the square root of 81?
Type: multiple_choice
9 (selected)
8
7
6

Figure 3.3.63. Update question set visibilities

- User clicks on the ban button inside a question set table row to complete ban a question set.
- By clicking the button again will unban the question set.

3.3.64 Administrator update group visibilities

The screenshot shows the Quizzzee platform's interface for managing groups. On the left, there's a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups (which is selected and highlighted in green), Reports, Subjects, and Fields. The main content area is titled "Group Testing" and displays a group named "Math please", created on May 3, 2025. The group has 1 member and 1 question set. The "Group Settings" section shows "Public" as Yes, "Can Invite" as Yes, and "Can Add Question Sets" as Yes. The "Group Stats" section shows 1 member and 1 question set. Below these, the "Group Leader" is listed as "Tory Tanner". A red box highlights the "BAN GROUP" button in the top right corner of the main content area.

Figure 3.3.64. Update group visibilities

- User clicks on the banning button inside a group table row to complete banning a group.
- By clicking the button again will unban the group.

3.3.65 View fields

3.3.65.1 Learner and guest view fields

The screenshot shows a user interface for a learning platform. At the top, there are navigation links: 'HOME' (green), 'EXPLORE' (red box), and a search bar. On the right, there are icons for language ('EN'), notifications ('50'), and profile ('User'). Below the navigation, a section titled 'Pick up where you left' contains three cards:

- Test Question Set #1**: 0% Not started. Created by 'thegaelord' (21 questions). Description: Explore fundamental concepts in mathematics, including algebra and geometry. Tags: Calculus, Linear Algebra, Statistics. Rating: 0.
- Test Question Set #3**: 0% Not started. Created by 'thegaelord' (6 questions). Description: Learn about the fundamental concepts of biology, including cell structure and genetics. Tags: General Biology, Molecular Biology, Ecology. Rating: 0.
- Test Question Set #2**: 67% Halfway done. Created by 'thegaelord' (15 questions). Description: Understand the basic principles of physics, from motion to energy. Tags: Classical Mechanics, Quantum Physics, Electromagnetism. Rating: 4.0.

At the bottom of the page, a reminder says: 'Remember to do your goals!'

Figure 3.3.65.1.a: View fields - 1

- User clicks on the explore page to view a list of fields

The screenshot shows the 'Explore' page with a header: 'Explore' and 'Categorize your goals by subject, allowing you to focus on specific areas'. Below the header, there are six categories:

- Biology**: Study of living organisms, their structure, function, growth, and evolution. Sub-topics: General Biology, Molecular Biology, Ecology.
- Mathematics**: Study of numbers, quantities, shapes, and patterns. Sub-topics: Calculus, Linear Algebra, Statistics.
- Physics**: Study of matter, energy, and the fundamental forces of nature. Sub-topics: Classical Mechanics, Quantum Physics, Electromagnetism.
- Chemistry**: Study of substances, their properties, structure, and the changes they undergo. Sub-topics: Organic Chemistry, Inorganic Chemistry, Biochemistry.
- History**
- Geography**

Figure 3.3.65.1.b: View fields - 2

3.3.65.2 Administrator views fields

The screenshot shows the Quizzzee application interface. On the left, there is a sidebar with various navigation options: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. The 'Fields' button is highlighted with a red box. The main content area is titled 'Fields' and displays a table with 10 fields found. The table has three columns: 'LABEL' (with a sorting arrow), 'Description', and 'Total Subjects'. Each row contains a field name, its description, and the number of subjects it covers. There is also a filter bar at the top and navigation buttons at the bottom.

LABEL	Description	Total Subjects
Biology	Study of living organisms, their structure, function, growth, and evolution	3
Mathematics	Study of numbers, quantities, shapes, and patterns	3
Physics	Study of matter, energy, and the fundamental forces of nature	3
Chemistry	Study of substances, their properties, structure, and the changes they undergo	3
History	Study of past events and societies	3
Geography	Study of places, natural features, and the distribution of life on Earth	3
Literature	Study of written works, especially those considered of superior or lasting artistic merit	3
Computer Science	Study of computation, data structures, and computer technology	3
Economics	Study of how societies use scarce resources to produce valuable goods and services	3
Language	Study of human communication through structured systems of symbols	3

Figure 3.3.65.2: View fields

- User clicks on the fields button to be redirected to the fields page.
- Inside the field page, the user can view a table of fields.

3.3.66 Administrator view field detail

The screenshot shows the 'Fields' section of the Quizzzee admin interface. On the left is a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. The main area is titled 'Fields' and shows a list of 10 fields found. A search bar labeled 'Filter Label...' is at the top. A table lists the fields with columns: LABEL, Description, and Total Subjects. The 'Biology' row is highlighted with a red border. The table rows are as follows:

LABEL	Description	Total Subjects
Biology	Study of living organisms, their structure, function, growth, and evolution	3
Mathematics	Study of numbers, quantities, shapes, and patterns	3
Physics	Study of matter, energy, and the fundamental forces of nature	3
Chemistry	Study of substances, their properties, structure, and the changes they undergo	3
History	Study of past events and societies	3
Geography	Study of places, natural features, and the distribution of life on Earth	3
Literature	Study of written works, especially those considered of superior or lasting artistic merit	3
Computer Science	Study of computation, data structures, and computer technology	3
Economics	Study of how societies use scarce resources to produce valuable goods and services	3
Language	Study of human communication through structured systems of symbols	3

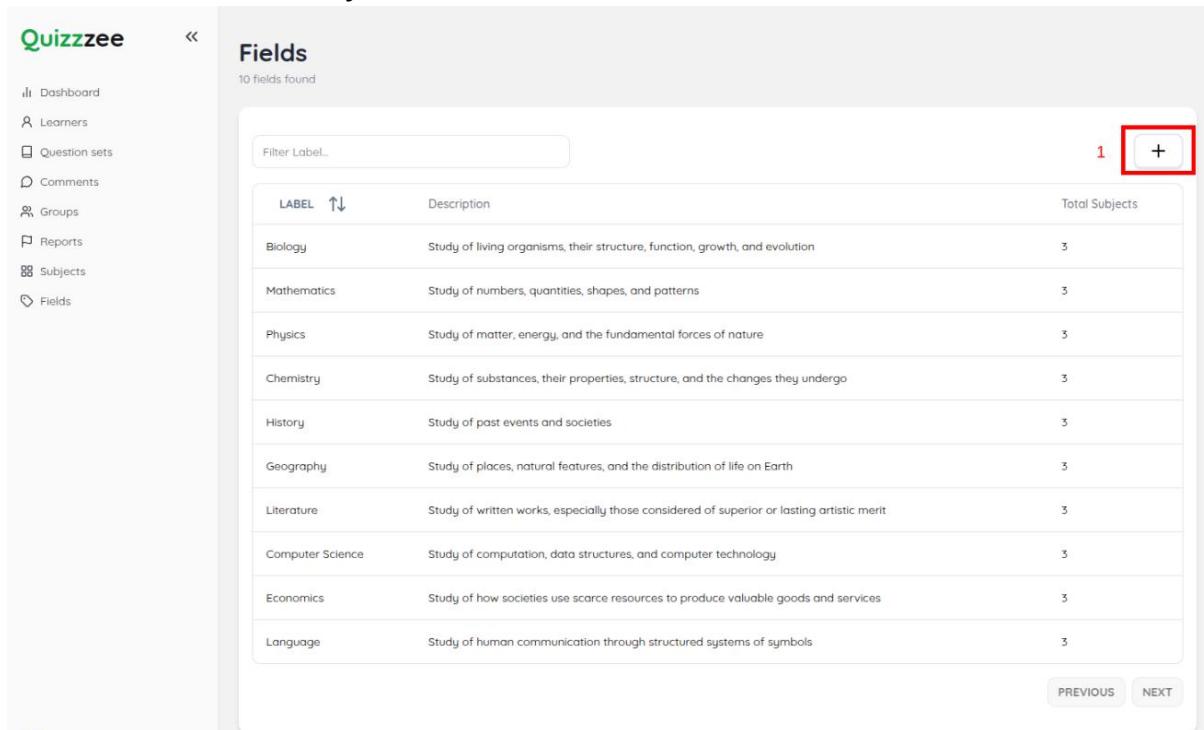
Figure 3.3.66.a: View field detail - 1

- User clicks on the field card to be redirected to the field detail page.

The screenshot shows the detailed view for the 'Biology' field. At the top, there is a preview card for 'Biology' (Sinh học) with an icon, an 'Update' button, and a 'Delete' button. Below this, there are two sections: 'English Description' and 'Vietnamese Description'. The English description is: 'Study of living organisms, their structure, function, growth, and evolution'. The Vietnamese description is: 'Nghiên cứu về sinh vật, cấu trúc, chức năng, sự phát triển và tiến hóa của chúng'. At the bottom, there is a 'Subjects' section with three categories: 'General Biology' (Sinh học đại cương), 'Molecular Biology' (Sinh học phân tử), and 'Ecology' (Sinh thái học). Each category has a brief description.

Figure 3.3.66.b: View field detail - 2

3.3.67 Administrator create fields



The screenshot shows the 'Fields' page in the Quizzzee application. On the left is a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. The main area is titled 'Fields' and displays a table with 10 rows. The table has columns for 'LABEL' (with a sorting arrow icon), 'Description', and 'Total Subjects' (which is always 3). The rows list various subjects: Biology, Mathematics, Physics, Chemistry, History, Geography, Literature, Computer Science, Economics, and Language. Each row contains a brief description of the subject. At the top right of the table, there is a red box highlighting a large blue '+' button. Below the table are 'PREVIOUS' and 'NEXT' navigation buttons.

LABEL	Description	Total Subjects
Biology	Study of living organisms, their structure, function, growth, and evolution	3
Mathematics	Study of numbers, quantities, shapes, and patterns	3
Physics	Study of matter, energy, and the fundamental forces of nature	3
Chemistry	Study of substances, their properties, structure, and the changes they undergo	3
History	Study of past events and societies	3
Geography	Study of places, natural features, and the distribution of life on Earth	3
Literature	Study of written works, especially those considered of superior or lasting artistic merit	3
Computer Science	Study of computation, data structures, and computer technology	3
Economics	Study of how societies use scarce resources to produce valuable goods and services	3
Language	Study of human communication through structured systems of symbols	3

Figure 3.3.67.a: Create fields - 1

- (1) User clicks on the create button to be redirected to the create field page.

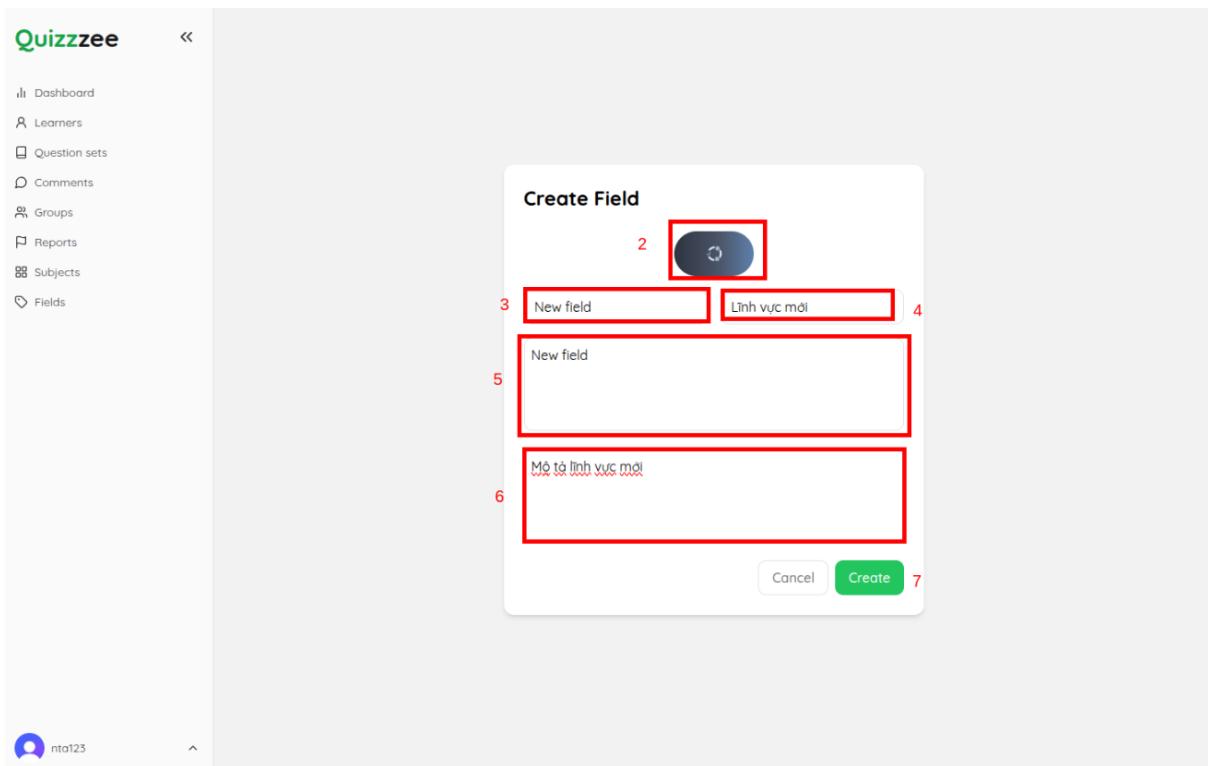


Figure 3.3.67.b: Create fields - 2

- (2) User chooses an image.
- (3) User inputs an English title.
- (4) User inputs a Vietnamese title.
- (5) User inputs an English description.
- (6) User inputs a Vietnamese description.
- (7) User clicks the create button to complete creating a subject.

3.3.68 Administrator update fields

The screenshot shows the Quizzzee platform interface. On the left, there's a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. The main area displays a subject card for 'Biology' (Sinh học). The card includes a biology icon, the subject name, and two buttons: 'Update' (highlighted with a red box) and 'Delete'. Below the card, there are sections for 'English Description' and 'Vietnamese Description', both describing the study of living organisms. At the bottom, there's a 'Subjects' section with three categories: General Biology, Molecular Biology, and Ecology, each with a brief description.

Figure 3.3.68.a: Update fields - 1

- (1) User clicks on the update icon inside the field table row to be redirected to the update field page.



Figure 3.3.68.b: Update field - 2

- (2) User chooses an image.
- (3) User inputs an English title.
- (4) User inputs an English description.
- (5) User inputs a Vietnamese title.
- (6) User inputs a Vietnamese description.
- (7) User clicks the update button to complete updating a subject.

3.3.69 Administrator delete fields

The screenshot shows the Quizzzee application interface. On the left, there is a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. The 'Subjects' link is currently selected. The main content area displays a subject card for 'Biology' (Sinh học). The card includes a 'Biology' icon, the title 'Biology' and 'Sinh học', an 'Update' button, and a 'Delete' button which is highlighted with a red box. Below the card, there are two sections: 'English Description' and 'Vietnamese Description'. The English description is 'Study of living organisms, their structure, function, growth, and evolution'. The Vietnamese description is 'Nghiên cứu về sinh vật, cấu trúc, chức năng, sự phát triển và tiến hóa của chúng'. At the bottom of the card, there is a 'Subjects' section with three sub-options: 'General Biology' (Sinh học đại cương), 'Molecular Biology' (Sinh học phân tử), and 'Ecology' (Sinh thái học). The user's profile icon ('nta123') is visible at the bottom left of the sidebar.

Figure 3.3.69: Delete fields

- User clicks on the delete icon to complete deleting a field.
- Only the Administrator can see the deleted fields.

3.3.70 View subjects

3.3.70.1 Learner and guest view subjects

The screenshot shows a user interface for a learning platform. At the top, there are navigation links: 'HOME' (green), 'EXPLORE' (red-bordered), 'Search with keyword' (with a magnifying glass icon), and other icons for language ('EN'), notifications ('50'), and profile ('User'). Below the header, a section titled 'Pick up where you left' has a small icon of a person sitting at a desk.

Three 'Test Question Set' cards are displayed:

- Test Question Set #1**: 'Not started' (0%). Description: 'Explore fundamental concepts in mathematics, including algebra and geometry.' Topics: Calculus, Linear Algebra, Statistics. Rating: 0 stars.
- Test Question Set #3**: 'Not started' (0%). Description: 'Learn about the fundamental concepts of biology, including cell structure and genetics.' Topics: General Biology, Molecular Biology, Ecology. Rating: 0 stars.
- Test Question Set #2**: 'Halfway done' (67%). Description: 'Understand the basic principles of physics, from motion to energy.' Topics: Classical Mechanics, Quantum Physics, Electromagnetism. Rating: 4.0 stars.

At the bottom of the page, a reminder is shown: 'Remember to do your goals!'.

Figure 3.3.70.1.a: View subjects - 1

- User clicks on the explore page to view a list of subjects

The screenshot shows the 'Explore' page with a search bar and various global settings at the top.

The main heading is 'Explore' with the sub-instruction: 'Categorize your goals by subject, allowing you to focus on specific areas'.

A grid of subject categories is displayed:

- Biology**: Study of living organisms, their structure, function, growth, and evolution. Sub-topics: General Biology, Molecular Biology, Ecology.
- Mathematics**: Study of numbers, quantities, shapes, and patterns. Sub-topics: Calculus, Linear Algebra, Statistics.
- Physics**: Study of matter, energy, and the fundamental forces of nature. Sub-topics: Classical Mechanics, Quantum Physics, Electromagnetism.
- Chemistry**: Study of substances, their properties, structure, and the changes they undergo. Sub-topics: Organic Chemistry, Inorganic Chemistry, Biochemistry.
- History**
- Geography**

Figure 3.3.70.1.b: View subjects - 2

3.3.70.2 Administrator views subjects

The screenshot shows the Quizzzee application interface. On the left, a sidebar menu includes options like Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects (which is highlighted with a red box), and Fields. The main content area is titled 'Subjects' and shows a table with 30 subjects found. The table has columns for 'LABEL' (with an up-down arrow icon), 'Description', and 'Field'. The subjects listed are: General Biology, Molecular Biology, Ecology, Calculus, Linear Algebra, Statistics, Classical Mechanics, Quantum Physics, Electromagnetism, and Organic Chemistry. Each row provides a brief description and its corresponding field (Biology, Mathematics, or Chemistry). At the bottom right of the table are 'PREVIOUS' and 'NEXT' buttons.

LABEL ↑	Description	Field
General Biology	Introduction to basic biological concepts, principles, and systems including cell structure, genetics, evolution, and physiology.	Biology
Molecular Biology	Study of biology at the molecular level focusing on DNA, RNA, proteins, and the mechanisms of genetic information transfer.	Biology
Ecology	Examination of organisms and their interactions with each other and the environment, including ecosystems, populations, and biodiversity.	Biology
Calculus	Study of continuous change and motion, including limits, derivatives, integrals, and infinite series.	Mathematics
Linear Algebra	Exploration of vector spaces, linear transformations, matrices, and systems of linear equations.	Mathematics
Statistics	Analysis of data collection, organization, analysis, interpretation and presentation of data.	Mathematics
Classical Mechanics	Study of motion of bodies under the influence of forces, including Newton's laws, kinematics, dynamics, and conservation principles.	Physics
Quantum Physics	Exploration of matter and energy at the atomic and subatomic levels, including wave-particle duality and quantum states.	Physics
Electromagnetism	Study of electrical and magnetic phenomena, Maxwell's equations, electromagnetic waves, and their applications.	Physics
Organic Chemistry	Study of carbon compounds, their structures, properties, reactions, and synthesis, with focus on hydrocarbons and their derivatives.	Chemistry

Figure 3.3.70.2: View subjects

- User clicks on the subjects button to be redirected to the subjects page.
- Inside the field page, the user can view a table of subjects.

3.3.71 Administrator view subject detail

The screenshot shows the Quizzzee application interface. On the left is a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects (which is currently selected), and Fields. The main area is titled 'Subjects' and displays a list of 30 subjects. A search bar labeled 'Filter Label...' is at the top. Below it is a table with columns: 'LABEL' (with an up-down arrow icon), 'Description', and 'Field'. The first row, 'General Biology', has its entire row highlighted with a red border. The 'General Biology' row contains the following data:

LABEL	Description	Field
General Biology	Introduction to basic biological concepts, principles, and systems including cell structure, genetics, evolution, and physiology.	Biology
Molecular Biology	Study of biology at the molecular level focusing on DNA, RNA, proteins, and the mechanisms of genetic information transfer.	Biology
Ecology	Examination of organisms and their interactions with each other and the environment, including ecosystems, populations, and biodiversity.	Biology
Calculus	Study of continuous change and motion, including limits, derivatives, integrals, and infinite series.	Mathematics
Linear Algebra	Exploration of vector spaces, linear transformations, matrices, and systems of linear equations.	Mathematics
Statistics	Analysis of data collection, organization, analysis, interpretation and presentation of data.	Mathematics
Classical Mechanics	Study of motion of bodies under the influence of forces, including Newton's laws, kinematics, dynamics, and conservation principles.	Physics
Quantum Physics	Exploration of matter and energy at the atomic and subatomic levels, including wave-particle duality and quantum states.	Physics
Electromagnetism	Study of electrical and magnetic phenomena, Maxwell's equations, electromagnetic waves, and their applications.	Physics
Organic Chemistry	Study of carbon compounds, their structures, properties, reactions, and synthesis, with focus on hydrocarbons and their derivatives.	Chemistry

At the bottom right of the table are 'PREVIOUS' and 'NEXT' buttons. The user's profile picture and name 'nta123' are visible at the bottom left.

Figure 3.3.71.a: View subject detail - 1

- User clicks on the subject card to be redirected to the subject detail page.

This screenshot shows the detailed view for the 'General Biology' subject. At the top, there is a small icon of a DNA helix and the text 'General Biology' followed by its Vietnamese translation 'Sinh học đại cương'. Below this are two buttons: 'Update' (blue) and 'Delete' (red).

The main content area is divided into two sections: 'English Description' and 'Vietnamese Description'.

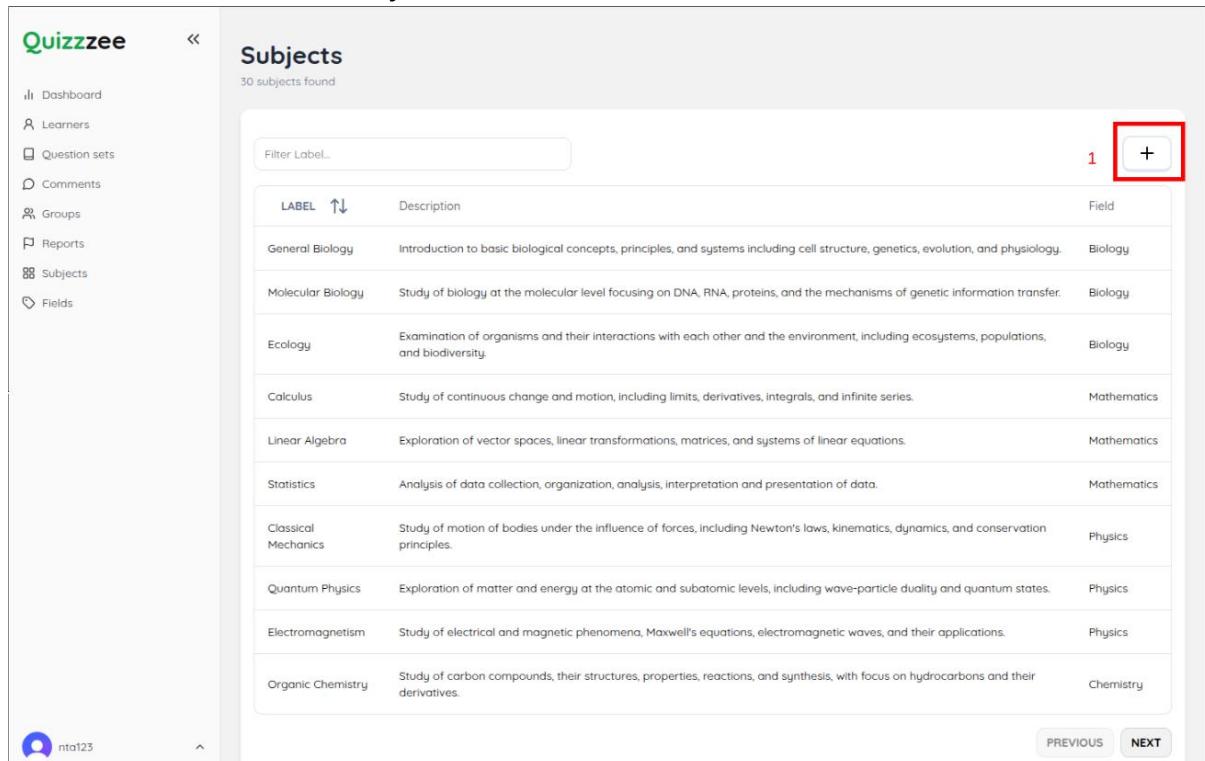
English Description:
Introduction to basic biological concepts, principles, and systems including cell structure, genetics, evolution, and physiology.

Vietnamese Description:
Giới thiệu về các khái niệm, nguyên tắc và hệ thống sinh học cơ bản bao gồm cấu trúc tế bào, di truyền học, tiến hóa và sinh lý học.

The user's profile picture and name 'nta123' are visible at the bottom left.

Figure 3.3.71.b: View subject detail - 2

3.3.72 Administrator create subjects



The screenshot shows the 'Subjects' page in the Quizzzee application. On the left is a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects (which is selected and highlighted in blue), and Fields. The main area has a title 'Subjects' with a back arrow and a note '30 subjects found'. Below is a table with columns 'LABEL' (with a sorting icon), 'Description', and 'Field'. The table contains nine rows of subject data. In the top right corner of the table header, there is a red box highlighting a '1' icon next to a '+' sign, indicating a new entry. At the bottom of the table are 'PREVIOUS' and 'NEXT' buttons.

LABEL	Description	Field
General Biology	Introduction to basic biological concepts, principles, and systems including cell structure, genetics, evolution, and physiology.	Biology
Molecular Biology	Study of biology at the molecular level focusing on DNA, RNA, proteins, and the mechanisms of genetic information transfer.	Biology
Ecology	Examination of organisms and their interactions with each other and the environment, including ecosystems, populations, and biodiversity.	Biology
Calculus	Study of continuous change and motion, including limits, derivatives, integrals, and infinite series.	Mathematics
Linear Algebra	Exploration of vector spaces, linear transformations, matrices, and systems of linear equations.	Mathematics
Statistics	Analysis of data collection, organization, analysis, interpretation and presentation of data.	Mathematics
Classical Mechanics	Study of motion of bodies under the influence of forces, including Newton's laws, kinematics, dynamics, and conservation principles.	Physics
Quantum Physics	Exploration of matter and energy at the atomic and subatomic levels, including wave-particle duality and quantum states.	Physics
Electromagnetism	Study of electrical and magnetic phenomena, Maxwell's equations, electromagnetic waves, and their applications.	Physics
Organic Chemistry	Study of carbon compounds, their structures, properties, reactions, and synthesis, with focus on hydrocarbons and their derivatives.	Chemistry

Figure 3.3.72.a: Create subjects - 1

- (1) User clicks on the create button to be redirected to the create field page.

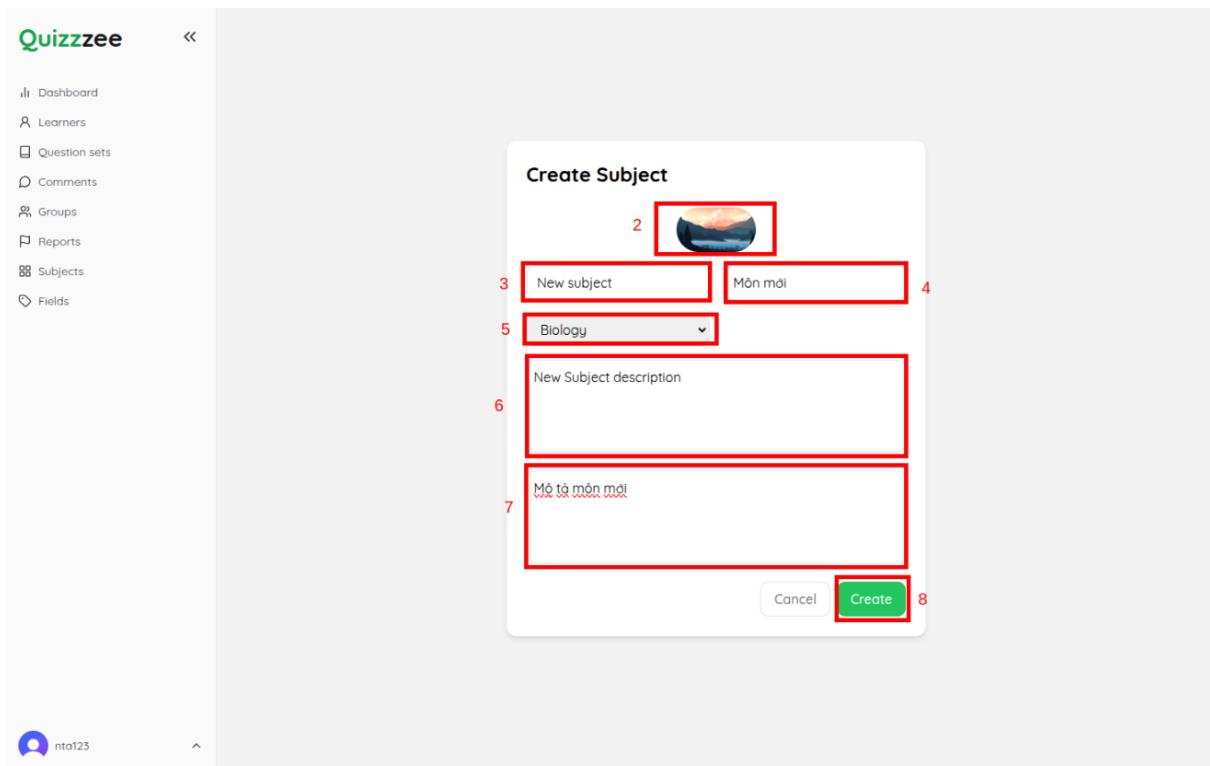


Figure 3.3.72.b: Create subjects - 2

- (2) User chooses an image.
- (3) User inputs an English title.
- (4) User inputs a Vietnamese title.
- (5) User chooses an associated field.
- (6) user input an English description.
- (7) User inputs a Vietnamese description.
- (8) User clicks the create button to complete creating a subject.

3.3.73 Administrator Update subjects

The screenshot shows the Quizzzee application interface. On the left, there is a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects (which is selected and highlighted in blue), and Fields. The main content area displays a subject detail page for "General Biology" (Sinh học đại cương). The page includes a thumbnail icon of a DNA helix, the subject name, and two buttons: "Update" (highlighted with a red box and the number 1) and "Delete". Below the subject name, there are two sections: "English Description" and "Vietnamese Description". The English description states: "Introduction to basic biological concepts, principles, and systems including cell structure, genetics, evolution, and physiology." The Vietnamese description states: "Giới thiệu về các khái niệm, nguyên tắc và hệ thống sinh học cơ bản bao gồm cấu trúc tế bào, di truyền học, tiến hóa và sinh lý học." At the bottom left of the main area, there is a user profile icon with the name "nta123".

Figure 3.3.73.a: Update subjects - 1

- (1) Inside the subject detail page, user clicks on the update button

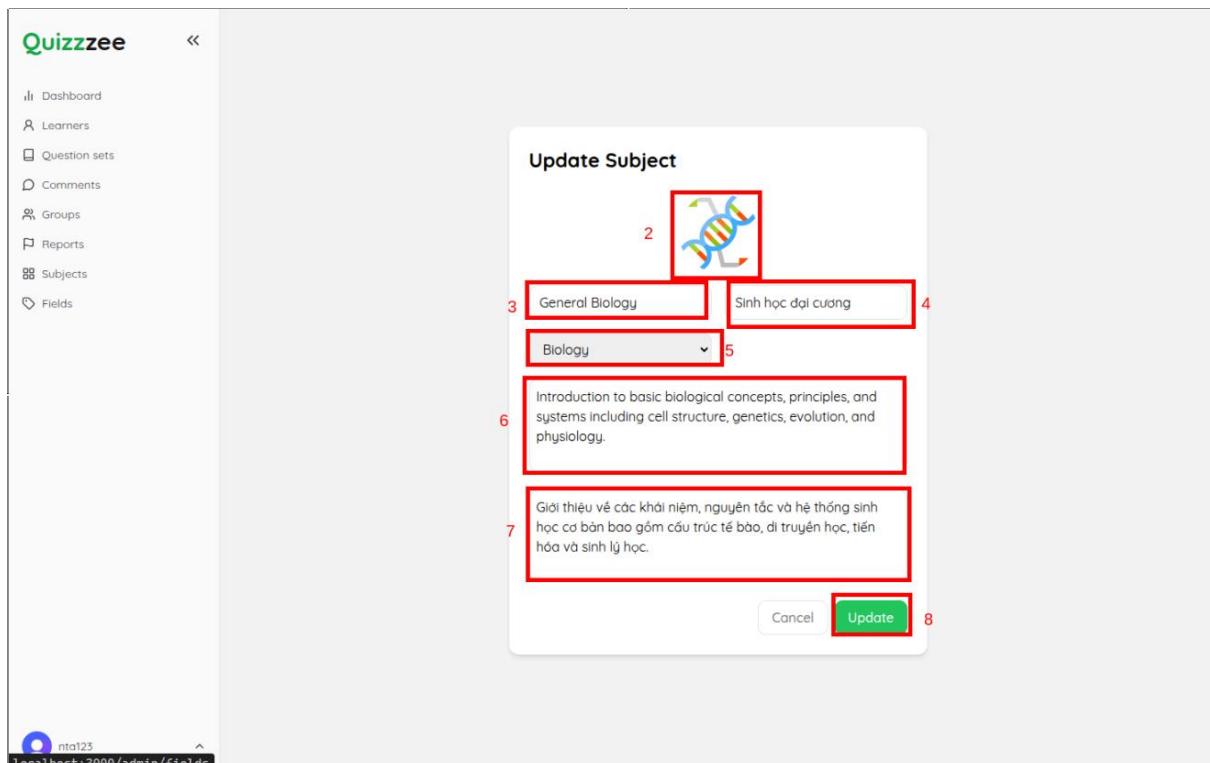


Figure 3.3.73.b: Update subjects - 2

- (2) User chooses an image.
- (3) User inputs an English title.
- (4) User inputs a Vietnamese title.
- (5) User chooses an associated field.
- (6) user input an English description.
- (7) User inputs a Vietnamese description.
- (8) User clicks the update button to complete updating a subject.

3.3.74 Administrator delete subjects

The screenshot shows the Quizzzee application interface. On the left, there is a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. The 'Subjects' link is highlighted. The main content area displays a subject card for 'General Biology' (Sinh học đại cương). The card features a DNA helix icon, the subject name in English and Vietnamese, and two buttons: 'Update' (blue) and 'Delete' (red, which is highlighted with a red box). Below the card, there are two sections: 'English Description' and 'Vietnamese Description'. The English description is: 'Introduction to basic biological concepts, principles, and systems including cell structure, genetics, evolution, and physiology.' The Vietnamese description is: 'Giới thiệu về các khái niệm, nguyên tắc và hệ thống sinh học cơ bản bao gồm cấu trúc tế bào, di truyền học, tiến hóa và sinh lý học.' At the bottom left of the main area, there is a user profile icon and the name 'nta123'.

Figure 3.3.74: Delete subjects

- User clicks on the delete button to complete deleting a subject.

3.3.75 Administrator view statistics

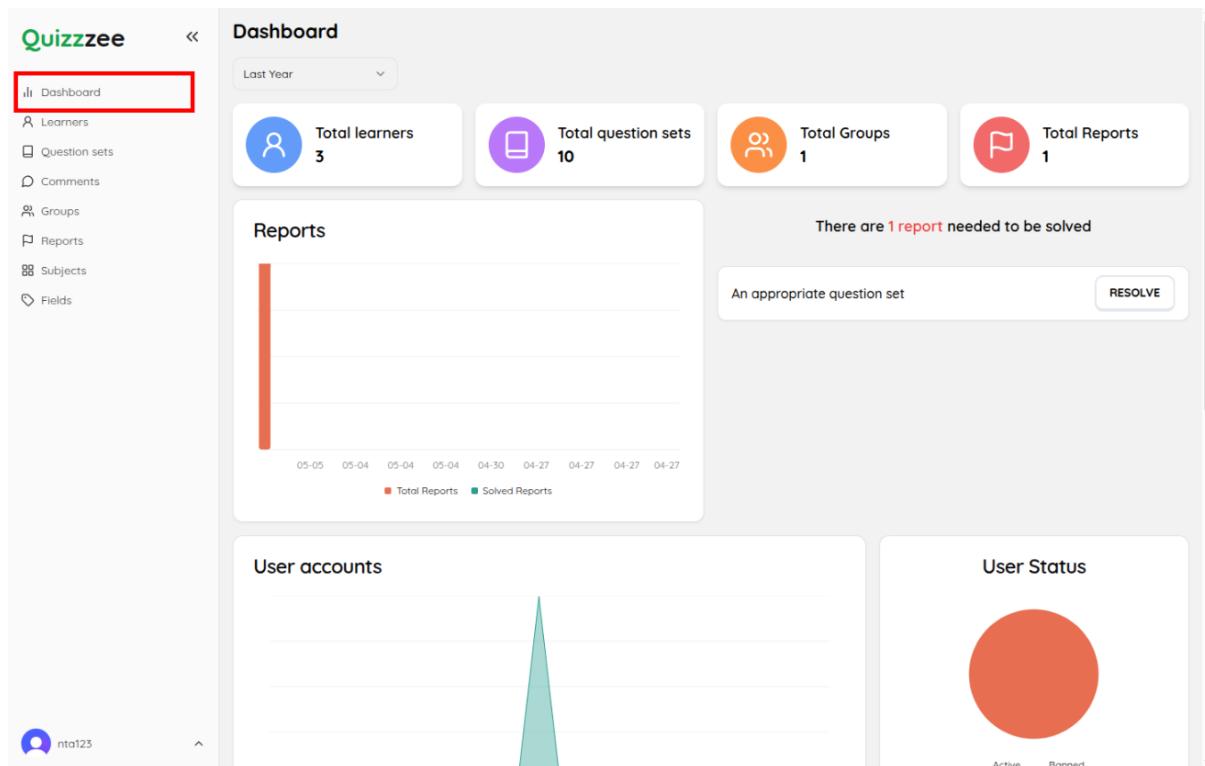


Figure 3.3.75 View statistics

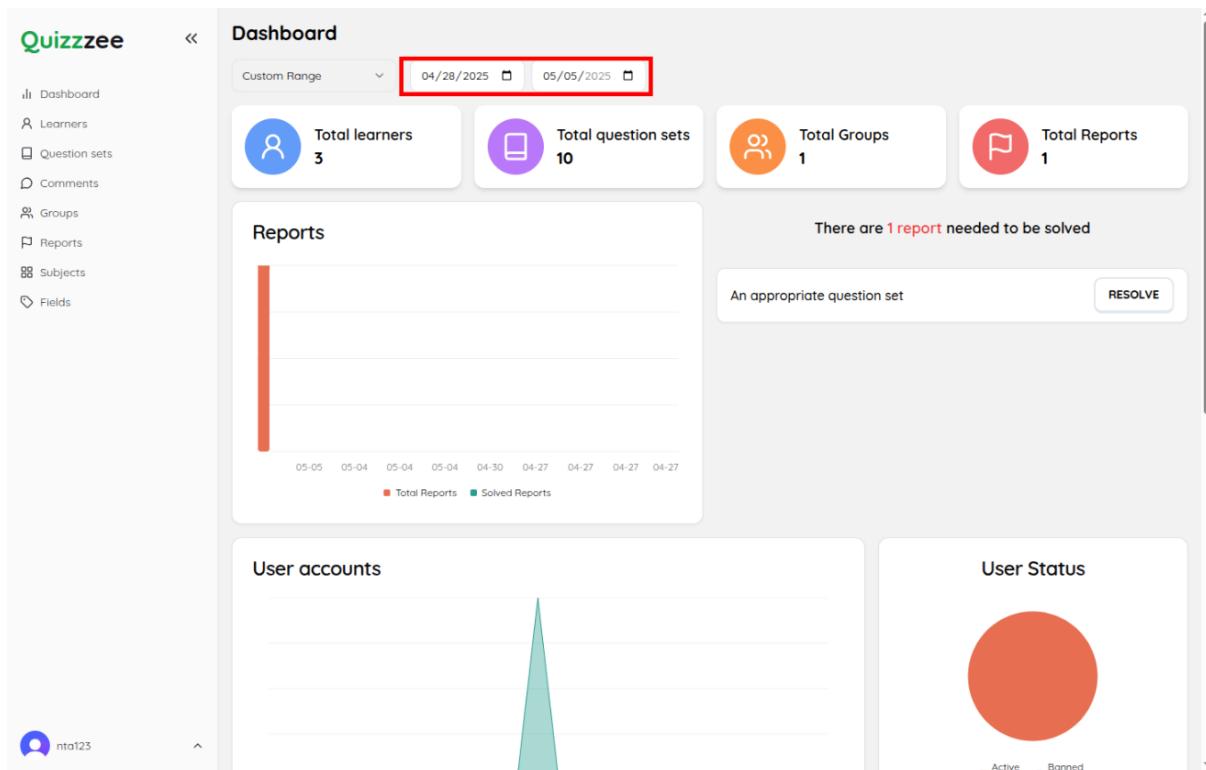
- User goes to the Dashboard page to view statistics.

3.3.76 Administrator filter statistics

The screenshot shows the Quizzzee administrator dashboard. On the left, there's a sidebar with navigation links: Dashboard, Learners, Question sets, Comments, Groups, Reports, Subjects, and Fields. The user is logged in as 'nta123'. The main area has a header 'Dashboard' with a back arrow. Below it is a date filter dropdown labeled 'Last Week' with a red box around it. A second red box highlights the 'Last Week' option in the dropdown menu. To the right of the dropdown are icons for 'Total question sets' (10), 'Total Groups' (1), and 'Total Reports' (1). A message says 'There are 1 report needed to be solved' with a 'RESOLVE' button. Below these are sections for 'Reports' (a bar chart showing 10 total reports) and 'User accounts' (empty). On the far right is a 'User Status' section with a large orange circle.

3.3.76.a: Filter statistics - 1

- (1) User clicks on date filter
- (2) User chooses an option to complete filtering statistics



3.3.76.b: Filter statistics - 2

- If the user chooses a custom date range, make sure to input the start and end date.

3.3.77 Learner and Guest use Pomodoro timer

The screenshot shows a digital learning platform interface. At the top, there are navigation links for 'HOME' and 'EXPLORE', a search bar, and various user account icons. Below this, a section titled 'Pick up where you left ⏸' lists three 'Test Question Set' items:

- Test Question Set #1**: 0% Not started. Description: Explore fundamental concepts in mathematics, including algebra and geometry. Topics: Calculus, Linear Algebra, Statistics. Rating: 0.
- Test Question Set #3**: 0% Not started. Description: Learn about the fundamental concepts of biology, including cell structure and genetics. Topics: General Biology, Molecular Biology, Ecology. Rating: 0.
- Test Question Set #2**: 67% Halfway done. Description: Understand the basic principles of physics, from motion to energy. Topics: Classical Mechanics, Quantum Physics, Electromagnetism. Rating: 4.0.

Below these sets, a reminder says 'Remember to do your goals!'. A goal card is shown with the text 'A goal' and 'Due date: May 8, 2025', with a status of 'Doing'. To the right, a red-bordered overlay displays a Pomodoro timer with a red circle containing a white stopwatch icon. The timer shows '1' above it, indicating the first work interval.

Figure 3.3.77.a: Use Pomodoro timer - 1

- (1) User clicks on the overlay timer.

This screenshot shows the same digital learning platform interface as the previous one, but with a modified Pomodoro timer overlay. The timer now displays '2' above the stopwatch icon, indicating the second work interval. The rest of the interface remains the same, including the three test question sets, the 'Remember to do your goals!' reminder, and the goal card.

Figure 3.3.77.b: Use Pomodoro timer - 2

- (2) User clicks on the option icon to modify timer (optional)

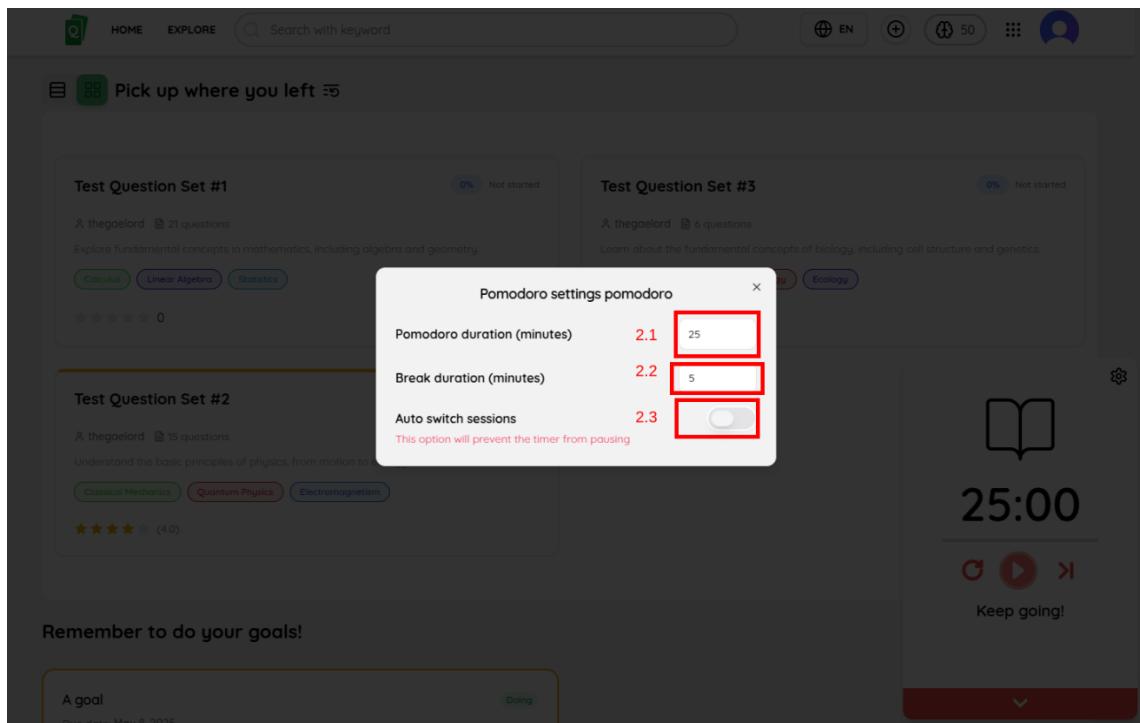


Figure 3.3.77.c: Use Pomodoro timer - 3

- (2.1) User update working time
- (2.2) User update rest time
- (2.3) User enable auto-switching session

Figure 3.3.77.c: Use Pomodoro timer - 3

- (3) User clicks the start button to start the timer.
- Once the timer reaches, it will make a sound to notify the user.
- (4) User clicks the next button to start rest time.
- If the current session is resting, then pressing the next button will switch to session mode.

Pick up where you left

Test Question Set #1 0% Not started
thegaelord 21 questions. Explore fundamental concepts in mathematics, including algebra and geometry. Calculus, Linear Algebra, Statistics. 0 stars.

Test Question Set #3 0% Not started
thegaelord 6 questions. Learn about the fundamental concepts of biology, including cell structure and genetics. General Biology, Molecular Biology, Ecology. 0 stars.

Test Question Set #2 67% Halfway done
thegaelord 15 questions. Understand the basic principles of physics, from motion to energy. Classical Mechanics, Quantum Physics, Electromagnetism. 4.0 stars.

Remember to do your goals!

A goal
Due date: May 8, 2025 Doing

Have some rest... 5 (C) 6 (P) 7 (R) 8 (F)

Figure 3.3.77.c: Use Pomodoro timer - 3

- (5) User clicks the reset button to reset the timer (optional)
- (6) User clicks the pause button to pause the timer. (optional)
- (7) User clicks the timer footer to hide the overlay (optional)
- If the user enables the auto-switching mode, the timer will automatically switch between working and resting sessions.

3.3.78 View achievements

3.3.78.1 Learner view achievements from achievements page

The screenshot shows the Achievements page on a learning platform. At the top right, there is a 'More' button (a grid icon) with a red box around it and the number '1'. Below it is a dropdown menu with options: 'My Groups', 'My Sets', 'Goals', and 'Achievements', with 'Achievements' also having a red box around it and the number '2'. The main content area displays three 'Test Question Set' cards:

- Test Question Set #1**: Not started (0%). Description: Explore fundamental concepts in mathematics, including algebra and geometry. Topics: Calculus, Linear Algebra, Statistics. Rating: 0.
- Test Question Set #3**: Not started (0%). Description: Learn about the fundamental concepts of biology, including cell structure and genetics. Topics: General Biology, Molecular Biology, Ecology. Rating: 0.
- Test Question Set #2**: Halfway done (67%). Description: Understand the basic principles of physics, from motion to energy. Topics: Classical Mechanics, Quantum Physics, Electromagnetism. Rating: 4.0.

At the bottom of the page, there is a grey bar with the text "Remember to do your goals!"

Figure 3.3.78.1.a View achievements - 1

- (1) User clicks on the more button
- (2) User clicks on the achievements button to be redirected to the achievements page.

The screenshot shows the Achievements page with a title 'Achievements' and a subtitle 'See your progress and milestones along your learning journey.' Below this, there is a 'Milestone' section. It features four items, each with a padlock icon and a progress bar at 0%:

- Neuron Igniter**: A grey card with a lock icon.
- Neuron Connector**: A grey card with a lock icon.
- Neuron Navigator**: A grey card with a lock icon.
- Neuron Commander**: A purple card with a lock icon.

A red box highlights the 'Neuron Commander' milestone.

Figure 3.3.78.1.b View achievements - 2

- Inside the achievements page, the user can see the current progress of each achievement.

3.3.78.3 Learner and Guest View achievements from learners profile page.

The screenshot shows a user profile page for 'Huynh Nguyen Gio Bao (K17 CT)'. At the top, there's a search bar and various navigation icons. On the left sidebar, under 'Achievements', there are several circular icons representing different achievements. The main content area displays four 'Test Question Set' boxes, each with a title, a brief description, and a category tag (e.g., 'World History', 'Ancient Civilizations', 'English', 'Modern European History', 'Data Structures and Algorithms').

Figure 3.3.78.3 View achievements

- Inside the learner profile page, user scrolls down until the achievements are displayed on the left sidebar.

3.3.79 View quotes

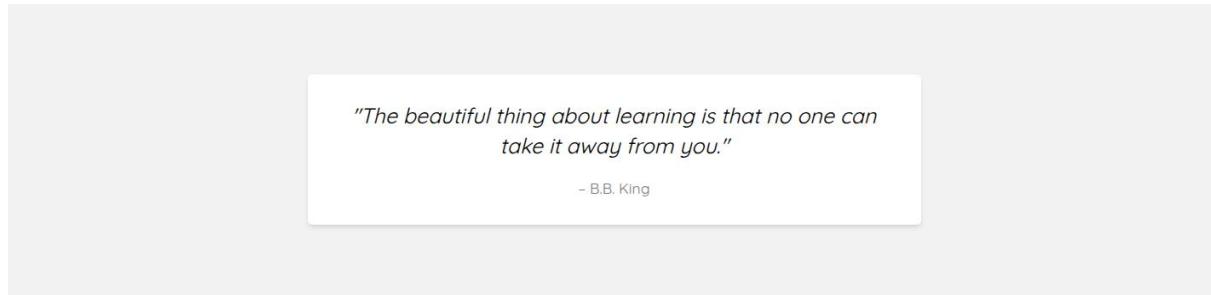


Figure 3.3.79. View quotes

- User navigates back to the Home page to view random quotes.
- By disabling quotes generation in the settings page, the quote will be hidden from the user.

VII. References

- [1] Clerk Official doc, available at: <https://clerk.com/docs>
- [2] Cloudinary Official doc, available at: <https://cloudinary.com/>
- [3] Ullman, Jeffrey D., and Jennifer Widom. "A First Course In Database Systems (3Rd)." (2022).
- [4] Wiegers, Karl E., and Joy Beatty. *Software requirements*. Pearson Education, 2013.
- [5] Crispin, Lisa, and Janet Gregory. *Agile testing: A practical guide for testers and agile teams*. Pearson Education, 2009.
- [6] Gamma, Erich. *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995.