

**HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY FOR HIGH QUALITY TRAINING**

****

**GRADUATE THESIS**

**BUILD A GRADUATE THESIS**

**MANAGEMENT SYSTEM**

**FOR FACULITY FOR HIGH QUALITY TRAINING**

**STUDENT NAME: STUDENT ID**

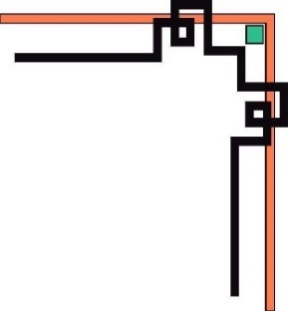
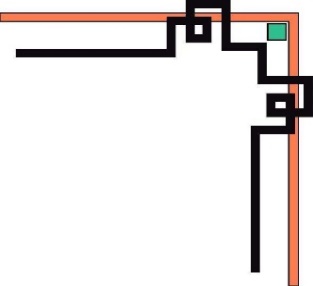
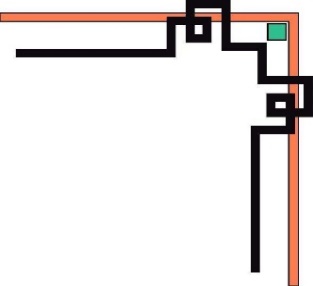
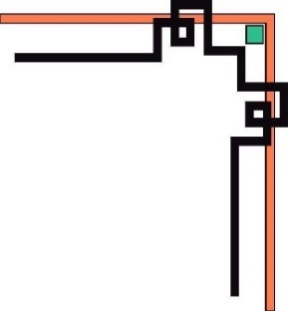
**NGUYEN DUY POON 16110186**

**School year: 2016 – 2020**

**Major: INFORMATION TECHNOLOGY**

**SUPERVISOR: M.Si LUONG VI MINH**

Ho Chi Minh, December 2020



**HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY FOR HIGH QUALITY TRAINING**

****

**GRADUATE THESIS**

**BUILD A GRADUATE THESIS**

**MANAGEMENT SYSTEM**

**FOR FACULITY FOR HIGH QUALITY TRAINING**

**STUDENT NAME: STUDENT ID**

**NGUYEN DUY POON 16110186**

**School year: 2016 – 2020**

**Major: INFORMATION TECHNOLOGY**

**SUPERVISOR: M.Si LUONG VI MINH**

Ho Chi Minh, December 2020

|  |  |
| --- | --- |
|  | **SOCIALIST REPUBLIC OF VIETNAM**  **Independence – Freedom - Happiness**  ----\*\*\*---- |

Ho Chi Minh, December 1st 2020

# MISSION OF GRADUATION THESIS

Student name Student ID Class

Nguyen Duy Poon 16110186 16110CLST4

Major: Information Technology

Mentor: M.Si Luong Vi Minh Contact: +84 35 621 6907

Started date: 14/09/2020 Submit date: 06/01/2021

1. Topic name: Build a graduate thesis management system for Faculity for High Quality Trainning.
2. Content to implement the project:

*Theory:*

* Research about the graduate thesis process of Faculity for High Quality Trainning.
* Research about Redis, MySQL and TypeORM.
* Research about RESTful APIs, NodeJS, Typescript, NestJS framework and Postman tool.
* Research about React, NextJS and Ant Design framework.
* Research about Docker Engine, Amazon S3, Heroku, Continues integration (CI) and GitHub Actions.

1. Products

* Graduate thesis process of Faculity for High Quality Trainning.
* RESTful APIs server.
* Graduate thesis application on web platform.

|  |  |
| --- | --- |
| HEAD OF INFORMATION TECHNOLOGY | SUPERVISOR |
| (Name and signature) | (Name and signature) |

# COMMENTARY OF SUPERVISOR

Student name: Nguyen Duy Poon Student ID: 16110186

Major: Information Technology.

Topic name: Build a graduate thesis management system for Faculity for High Quality Trainning.

Name of supervisor: M.Si Luong Vi Minh.

**COMMENTARY**

1. On content of topic & workload done:

1. Advantage:

1. Disadvantage:

1. Recommend for defense or not?
2. Rating type:
3. Mark: (By word: )

*Ho Chi Minh, ……………………………….* 2020

SUPERVISOR

(Name and signature)

|  |  |
| --- | --- |
|  | **SOCIALIST REPUBLIC OF VIETNAM**  **Independence – Freedom - Happiness**  ----\*\*\*---- |

# COMMENTARY OF REVIEWER

Student name: Nguyen Duy Poon Student ID: 16110186

Major: Information Technology.

Topic name: Build a graduate thesis management system for Faculity for High Quality Trainning.

Name of reviewer:

**COMMENTARY**

1. On content of topic & workload done:

1. Advantage:

1. Disadvantage:

1. Recommend for defense or not?
2. Rating type:
3. Mark: (By word: )

*Ho Chi Minh, ……………………………….* 2020

REVIEWER

(Name and signature)

# ASSURANCE

We assure that this project is our own implementation. We do not copy, use any material or source code of others without specifying the source. We assume responsibility for violations.

*Ho Chi Minh,* December 1st 2020

Nguyen Duy Poon

# MANY THANKS

Success doesn't come from one person. To complete this thesis, the author would like to sincerely thanks M.Si Luong Vi Minh who has supported me during this thesis processing. The project will not complete without his comments, guides, and practices. I'm very respectful and grateful for his effort. Once again, thank you very much.

Because of limited time and knowledge with many other reasons, so there will have inevitable problems, so I hope you feel free to raise your idea with me for more complete later. We sincerely thank you.

Ho Chi Minh, December 1st 2020

Nguyen Duy Poon

# SUMMARY INFORMATION BY VIETNAMESE

1. **Các vấn đề nghiên cứu**

* Tìm hiểu về quy trình tổ chức khóa luận của khoa Đào tạo Chất lượng cao.
* Tìm hiểu về công nghệ lưu trữ dữ liệu Redis, MySQL và công nghệ ánh xạ đối tượng quan hệ TypeORM.
* Tìm hiểu về các công nghệ để xây dựng máy chủ RESTful APIs như NodeJS, NestJS framework, Typescript giúp giải quyết rào cản về nền tảng hỗ trợ của ứng dụng công nghệ thông tin.
* Tìm hiểu về các công nghệ để xây dựng ứng dụng web hiện nay như React, NextJS framework và Ant Design framework.
* Tìm hiểu về Docker Engine, Tích hợp liên tục (CI), GitHub Actions, Amazon S3, Heroku và Postman trong việc phát triển, bảo trì và vận hành sản phẩm công nghệ thông tin.

1. **Kết quả đạt được**

* Quy trình tổ chức khóa luận của khoa Đào tạo Chất lượng cao.
* Máy chủ RESTfuls API.
* Hệ thống tổ chức khóa luận cho khoa Đào tạo Chất lượng cao trên nền tảng web.

# SUMMARY INFORMATION BY ENGLISH

1. **Research issues**

* Research about the graduate thesis process of Faculity for High Quality Trainning.
* Research about storage technologies such as Redis, MySQL and Object Relational Mapping (ORM) technology TypeORM.
* Research about building RESTful APIs server technologies such as NodeJS, NestJS framework and Typescript which help solve barriers to cross-platform support of the application of information technology.
* Research about building web application technologies such as React, NextJS framework and Ant Design framework.
* Research about Docker Engine, Continues Integration (CI), GitHub Actions, Amazon S3, Heroku and Postman in development, maintenance and deploy information technology product.

1. **Achieved results**

* Graduate thesis process of Faculity for High Quality Trainning.
* RESTful APIs server.
* Graduate thesis application on web platform.

# TABLE OF CONTENT

[MISSION OF GRADUATION THESIS i](#_Toc59042080)

[COMMENTARY OF SUPERVISOR ii](#_Toc59042081)

[COMMENTARY OF REVIEWER iii](#_Toc59042082)

[ASSURANCE iv](#_Toc59042083)

[MANY THANKS v](#_Toc59042084)

[SUMMARY INFORMATION BY VIETNAMESE vi](#_Toc59042085)

[SUMMARY INFORMATION BY ENGLISH vii](#_Toc59042086)

[TABLE OF CONTENT viii](#_Toc59042087)

[LIST OF TABLES xii](#_Toc59042088)

[LIST OF FIGURES xv](#_Toc59042089)

[CHAPTER 1. OVEWVIEW 18](#_Toc59042090)

[1.1. Urgency and objectives of project 18](#_Toc59042091)

[1.1.1. The urgency of project 18](#_Toc59042092)

[1.1.2. Project objectives 18](#_Toc59042093)

[1.2. Objects, scope and methods of research 18](#_Toc59042094)

[1.2.1. Research objects 18](#_Toc59042095)

[1.2.2. Research scope 19](#_Toc59042096)

[1.2.3. Research methods 19](#_Toc59042097)

[1.3. Scientific and practical meaning of the project 20](#_Toc59042098)

[CHAPTER 2. THEORETICAL BASIS 21](#_Toc59042099)

[2.1. The current graduate thesis process of Faculty of High Quality Training 21](#_Toc59042100)

[2.1.1. Participants and roles 21](#_Toc59042101)

[2.1.2. Phases 21](#_Toc59042102)

[2.2. Overview of MySQL 24](#_Toc59042103)

[2.2.1. Introduction [1] 24](#_Toc59042104)

[2.2.2. Features [1] 25](#_Toc59042105)

[2.2.3. History [1] 25](#_Toc59042106)

[2.2.4. Advantages [2] 25](#_Toc59042107)

[2.2.5. MySQL in project 26](#_Toc59042108)

[2.3. Overview of Redis 26](#_Toc59042109)

[2.3.1. Introduction [3] 26](#_Toc59042110)

[2.3.2. Features [3] 26](#_Toc59042111)

[2.3.3. History [4] 27](#_Toc59042112)

[2.3.4. Advantages [5] 27](#_Toc59042113)

[2.3.5. Redis in project 27](#_Toc59042114)

[2.4. Overview of Object-relational mapping (ORM) and TypeORM 28](#_Toc59042115)

[2.4.1. Object-relational mapping (ORM) [6] 28](#_Toc59042116)

[2.4.2. TypeORM [7] 28](#_Toc59042117)

[2.4.3. TypeORM in project 28](#_Toc59042118)

[2.5. Overview of NodeJS 29](#_Toc59042119)

[2.5.1. Introduction [8] 29](#_Toc59042120)

[2.5.2. Features [8] 29](#_Toc59042121)

[2.5.3. History [9] 29](#_Toc59042122)

[2.5.4. NodeJS in project 30](#_Toc59042123)

[2.6. Overview of TypeScript 30](#_Toc59042124)

[2.6.1. Introduction [10] 31](#_Toc59042125)

[2.6.2. Features [10] 31](#_Toc59042126)

[2.6.3. History [11] 31](#_Toc59042127)

[2.6.4. TypeScript in project 32](#_Toc59042128)

[2.7. Overview of NestJS 32](#_Toc59042129)

[2.7.1. Introduction [12] 32](#_Toc59042130)

[2.7.2. Features [12] 33](#_Toc59042131)

[2.7.3. NestJS in project 33](#_Toc59042132)

[2.8. Overview of React 33](#_Toc59042133)

[2.8.1. Introduction [13] 33](#_Toc59042134)

[2.8.2. Features [13] 34](#_Toc59042135)

[2.8.3. Advantages [13] 34](#_Toc59042136)

[2.8.4. History [14] 34](#_Toc59042137)

[2.8.5. React in project 35](#_Toc59042138)

[2.9. Overview of NextJS 35](#_Toc59042139)

[2.9.1. Introduction [15] 35](#_Toc59042140)

[2.9.2. Features [16] 36](#_Toc59042141)

[2.9.3. NextJS in project 36](#_Toc59042142)

[2.10. Overview of Ant Design (for React) 37](#_Toc59042143)

[2.10.1. Introduction [17] 37](#_Toc59042144)

[2.10.2. Features [17] 37](#_Toc59042145)

[2.10.3. Ant Design in project 37](#_Toc59042146)

[2.11. Other technologies and libraries 37](#_Toc59042147)

[2.11.1. Docker Engine [18] 37](#_Toc59042148)

[2.11.2. Continues Integration (CI) [19] and GitHub Actions [20] 38](#_Toc59042149)

[2.11.3. Amazon S3 [21] 40](#_Toc59042150)

[2.11.4. Heroku [22] 40](#_Toc59042151)

[2.11.1. Postman [23] 41](#_Toc59042152)

[CHAPTER 3. REQUIREMENT MODELING 42](#_Toc59042153)

[3.1. Define usecase 42](#_Toc59042154)

[3.1.1. Actor 42](#_Toc59042155)

[3.1.2. Use case 43](#_Toc59042156)

[3.2. Use case diagram 46](#_Toc59042157)

[3.3. Use case specification 46](#_Toc59042158)

[3.3.1. Manage lecturers 46](#_Toc59042159)

[3.3.2. Manage students 49](#_Toc59042160)

[3.3.3. Manage theses 53](#_Toc59042161)

[3.3.4. Manage defense councils 56](#_Toc59042162)

[3.3.5. Manage topics 60](#_Toc59042163)

[3.3.6. Process topic approval request 65](#_Toc59042164)

[3.3.7. Process topic register request 69](#_Toc59042165)

[3.3.8. Edit review information 72](#_Toc59042166)

[3.3.9. Edit defense information 72](#_Toc59042167)

[CHAPTER 4. SOFTWARE DESIGN 74](#_Toc59042168)

[4.1. System design 74](#_Toc59042169)

[4.1.1. Workflow 74](#_Toc59042170)

[4.2. Database design 74](#_Toc59042171)

[4.2.1. Database diagram 74](#_Toc59042172)

[4.2.2. Description for each table 74](#_Toc59042173)

[4.3. Interface design 85](#_Toc59042174)

[4.3.1. Screen flow 85](#_Toc59042175)

[4.3.2. Specification of the screens 85](#_Toc59042176)

[CHAPTER 5. INSTALLATION AND TESTING 125](#_Toc59042177)

[5.1. Installation 125](#_Toc59042178)

[5.1.1. Production environment 125](#_Toc59042179)

[5.1.2. Development environment 126](#_Toc59042180)

[5.2. Testing 128](#_Toc59042181)

[5.2.1. Thesis 128](#_Toc59042182)

[5.2.2. Topic 129](#_Toc59042183)

[CHAPTER 6. CONCLUSIONS AND DEVELOPMENT STRATEGY 134](#_Toc59042184)

[6.1. Results 134](#_Toc59042185)

[6.3. Restrictions 135](#_Toc59042186)

[6.4. Development strategy 135](#_Toc59042187)

[REFERENCES 136](#_Toc59042188)

[APPENDIX 1: DATABASE DIAGRAM 138](#_Toc59042189)

[APPENDIX 2: SCREEN FLOW 139](#_Toc59042190)

# LIST OF TABLES

[Table 2.1. Table evaluate criteria of the defense council 24](#_Toc59041723)

[Table 3.1. Actor 42](#_Toc59041724)

[Table 3.2. Use case 43](#_Toc59041725)

[Table 3.3. Specific view lecturers 47](#_Toc59041726)

[Table 3.4. Specific create lecturer 47](#_Toc59041727)

[Table 3.5. Specific edit lecturer 48](#_Toc59041728)

[Table 3.6. Specific delete lecturer 48](#_Toc59041729)

[Table 3.7. Specific search lecturer 49](#_Toc59041730)

[Table 3.8. Specific view students 50](#_Toc59041731)

[Table 3.9. Specific create student 50](#_Toc59041732)

[Table 3.10. Specific edit student 51](#_Toc59041733)

[Table 3.11. Specific delete student 52](#_Toc59041734)

[Table 3.12. Specific search student 52](#_Toc59041735)

[Table 3.13. Specific create thesis 53](#_Toc59041736)

[Table 3.14. Specific edit thesis 54](#_Toc59041737)

[Table 3.15. Specific delete thesis 55](#_Toc59041738)

[Table 3.16. Specific active thesis 55](#_Toc59041739)

[Table 3.17. Specific inactive thesis 56](#_Toc59041740)

[Table 3.18. Specific view defense councils 57](#_Toc59041741)

[Table 3.19. Specific create defense council 58](#_Toc59041742)

[Table 3.20. Specific edit defense council 58](#_Toc59041743)

[Table 3.21. Specific delete defense council 59](#_Toc59041744)

[Table 3.22. Specific search defense council 60](#_Toc59041745)

[Table 3.23. Specific create topic 61](#_Toc59041746)

[Table 3.24. Specific edit topic 62](#_Toc59041747)

[Table 3.25. Specific delete topic 63](#_Toc59041748)

[Table 3.26. Specific cancel topic 63](#_Toc59041749)

[Table 3.27. Specific open register 64](#_Toc59041750)

[Table 3.28. Specific close register 65](#_Toc59041751)

[Table 3.29. Specific send approval request 66](#_Toc59041752)

[Table 3.30. Specific withdraw approval request 67](#_Toc59041753)

[Table 3.31. Specific accept approval request 67](#_Toc59041754)

[Table 3.32. Specific reject approval request 68](#_Toc59041755)

[Table 3.33. Specific sendback approval request 69](#_Toc59041756)

[Table 3.34. Specific send register request 70](#_Toc59041757)

[Table 3.35. Specific accept register request 70](#_Toc59041758)

[Table 3.36. Specific reject register request 71](#_Toc59041759)

[Table 3.37. Specific edit review information 72](#_Toc59041760)

[Table 3.38. Specific edit defense information 73](#_Toc59041761)

[Table 4.1. Description for user table 74](#_Toc59041762)

[Table 4.2. Description for user\_refresh\_token table 75](#_Toc59041763)

[Table 4.3. Description for lecturer table 76](#_Toc59041764)

[Table 4.4. Description for student table 76](#_Toc59041765)

[Table 4.5. Description for thesis table 77](#_Toc59041766)

[Table 4.6. Description for thesis\_lecturer table 78](#_Toc59041767)

[Table 4.7. Description for thesis\_student table 78](#_Toc59041768)

[Table 4.8. Description for topic table 79](#_Toc59041769)

[Table 4.9. Description for topic\_state table 79](#_Toc59041770)

[Table 4.10. Description for topic\_student table 80](#_Toc59041771)

[Table 4.11. Description for progress\_report table 80](#_Toc59041772)

[Table 4.12. Description for review table 81](#_Toc59041773)

[Table 4.13. Description for council table 82](#_Toc59041774)

[Table 4.14. Description for defense table 82](#_Toc59041775)

[Table 4.15. Description for result table 83](#_Toc59041776)

[Table 4.16. Description for comment table 84](#_Toc59041777)

[Table 4.17. Description for migrations table 84](#_Toc59041778)

[Table 4.18. Thesis list screen specification 85](#_Toc59041779)

[Table 4.19. Create thesis screen specification 86](#_Toc59041780)

[Table 4.20. Thesis list tabscreen specification 90](#_Toc59041781)

[Table 4.21. Instructor list tabscreen specification 92](#_Toc59041782)

[Table 4.22. Student list tabscreen specification 93](#_Toc59041783)

[Table 4.23. Topic list tab specification 95](#_Toc59041784)

[Table 4.24. Council list tab specification 97](#_Toc59041785)

[Table 4.25. Topic detail tab specification 100](#_Toc59041786)

[Table 4.26. Approval information tab specification 103](#_Toc59041787)

[Table 4.27. Register list tab specification 106](#_Toc59041788)

[Table 4.28. Semi-progress report tab specification 110](#_Toc59041789)

[Table 4.29. Review tab specification 116](#_Toc59041790)

[Table 4.30. Defense tab specification 120](#_Toc59041791)

[Table 4.31. Result tab specification 123](#_Toc59041792)

[Table 5.1. Libraries and software need to be installed in production 125](#_Toc59041793)

[Table 5.2. Libraries and software need to be installed in development 126](#_Toc59041794)

[Table 5.3. List of command for install and run projects 126](#_Toc59041795)

[Table 5.4. Test case function create thesis 128](#_Toc59041796)

[Table 5.5. Test case function active thesis 129](#_Toc59041797)

[Table 5.6. Test case function create topic 130](#_Toc59041798)

[Table 5.7. Test case function approval topic 131](#_Toc59041799)

[Table 5.8. Test case function upload report document 132](#_Toc59041800)

# LIST OF FIGURES

[Figure 2.2.1. Logo of MySQL 25](#_Toc59041801)

[Figure 2.2. Logo of TypeORM 28](#_Toc59041802)

[Figure 2.3. Logo of NodeJS 29](#_Toc59041803)

[Figure 2.4. Logo of TypeScript 31](#_Toc59041804)

[Figure 2.5. Logo of NestJS 33](#_Toc59041805)

[Figure 2.6. Logo of React 34](#_Toc59041806)

[Figure 2.7. Logo of NextJS 36](#_Toc59041807)

[Figure 2.8. Logo of Ant Design 37](#_Toc59041808)

[Figure 2.9. Logo of Docker 38](#_Toc59041809)

[Figure 2.10. Continuous Integration life cycle 39](#_Toc59041810)

[Figure 2.11. Logo of GitHub Actions 39](#_Toc59041811)

[Figure 2.12. Logo of Heroku 41](#_Toc59041812)

[Figure 2.13. Logo of Postman 41](#_Toc59041813)

[Figure 3.1. Use case diagram 46](https://d.docs.live.net/499da2eccd05aa12/KLTN2020-2/kltn-16110186.docx#_Toc59041814)

[Figure 3.2. Use case manage lecturers 47](#_Toc59041815)

[Figure 3.3. Use case manage students 50](#_Toc59041816)

[Figure 3.4. Use case manage theses 53](#_Toc59041817)

[Figure 3.5. Use case manage defense councils 57](#_Toc59041818)

[Figure 3.6. Use case manage topics 61](#_Toc59041819)

[Figure 3.7. Use case process topic approval request 66](#_Toc59041820)

[Figure 3.8. Use case process topic register request 69](#_Toc59041821)

[Figure 3.9. Use case edit review information 72](#_Toc59041822)

[Figure 3.10. Use case edit defense information 73](#_Toc59041823)

[Figure 4.1. Workflow about graduate thesis 74](#_Toc59041824)

[Figure 4.2. Thesis list screen 85](#_Toc59041825)

[Figure 4.3. Create thesis screen 86](#_Toc59041826)

[Figure 4.4. Thesis detail screen 89](#_Toc59041827)

[Figure 4.5. Delete thesis confirm dialog 89](#_Toc59041828)

[Figure 4.6. Active thesis confirm dialog 90](#_Toc59041829)

[Figure 4.7. Inactive thesis confirm dialog 90](#_Toc59041830)

[Figure 4.8. Instructor list tab 92](#_Toc59041831)

[Figure 4.9. Student list tab 93](#_Toc59041832)

[Figure 4.10. Topic list tab 94](#_Toc59041833)

[Figure 4.11. Create topic drawer 95](#_Toc59041834)

[Figure 4.12. Council list tab 96](#_Toc59041835)

[Figure 4.13. Create council 96](#_Toc59041836)

[Figure 4.14. Delete council confirm dialog 97](#_Toc59041837)

[Figure 4.15. Detail topic tab 99](#_Toc59041838)

[Figure 4.16. Change resiter status of topic confirm dialog 100](#_Toc59041839)

[Figure 4.17. Approval information tab 103](#_Toc59041840)

[Figure 4.18. Confirm action dialog 103](#_Toc59041841)

[Figure 4.19. Register list tab 105](#_Toc59041842)

[Figure 4.20. Register confirm dialog 106](#_Toc59041843)

[Figure 4.21. Register confirm dialog 106](#_Toc59041844)

[Figure 4.22. Semi-progress report tab 109](#_Toc59041845)

[Figure 4.23. Edit semi-progress information drawer 109](#_Toc59041846)

[Figure 4.24. Change result confirm dialog 110](#_Toc59041847)

[Figure 4.25. Delete comment confirm dialog 110](#_Toc59041848)

[Figure 4.25. Delete document report confirm dialog 110](#_Toc59041849)

[Figure 4.27. Review tab 114](#_Toc59041850)

[Figure 4.28. Edit review drawer 115](#_Toc59041851)

[Figure 4.29. Change result confirm drawer 116](#_Toc59041852)

[Figure 4.30. Change result confirm dialog 116](#_Toc59041853)

[Figure 4.31. Defense tab 119](#_Toc59041854)

[Figure 4.32. Edit defense drawer 120](#_Toc59041855)

[Figure 4.33. Result tab 122](#_Toc59041856)

[Figure 4.34. Edit result drawer 123](#_Toc59041857)

# CHAPTER 1. OVEWVIEW

## Urgency and objectives of project

### The urgency of project

The graduate thesis is one of the most important parts of a student's life. At some universities, it became a required condition for graduation. Besides, it also is considered that achievement or something like a memory of university time.

But the current graduate thesis process of Faculty for High Quality Training has inadequacies. The traditional process is mostly manual or semi-computerized through social networks such as Facebook. Therefore, it makes some difficult to manage, tracking and statistics. Besides, the information about the graduate thesis between attendees can be missing because don't have general communication.

With the development of information technology, every working processes are converted to computerized to optimizing resource and human force. No exceptions, the current graduate thesis process need to replace the traditional process with a computerized process to improve the above inadequacies.

As an information technology student, the author more understands than anyone else the necessity of information technology in the graduate thesis process of Faculty for High Quality Trainning. Therefore, with the agreement of Head of Infomation Technology Nguyen Dang Quang, the author choose the topic "Build a graduate thesis management system for Faculty for High Quality Training" for project of graduate thesis.

### Project objectives

Topic “Build a graduate thesis management system for Faculty for High Quality Training” is solve the issues include:

* Understand the graduate thesis process of Faculty for High Quality Training, including attendee type (with permission), the states of a graduate thesis, the allowed behaviors of attendee with each state.
* Use the above process to build a graduate thesis management system for computerized the graduate thesis process, including student management, lecturer management, graduate thesis management (with graduate thesis state management).

## Objects, scope and methods of research

### Research objects

The research objects of this graduate thesis include two main objects:

* The current graduate thesis process of Faculty for High Quality Training.
* The technologies apply to build a graduate thesis management system on the web platform. In this object, it includes smaller research objects which are separated by the following layers:
  + Database layer: Redis database, MySQL database, Object Relation Mapping technology TypeORM.
  + API server layer: NodeJS platform, Typescript language, NestJS framework and RESTful APIs.
* Web application layer: React library, NextJS framework and Ant Design framework.
* Development and deploy layer: Docker Engine, Continues Integration (CI), GitHub Actions, Amazon S3, Heroku and Postman.

### Research scope

The research scope of this thesis is limited by the current graduate thesis process of Faculty for High Quality Training with the conclusion from experimental research results from the author during this project implementation.

### Research methods

To implement this project, the author was used the mixed research method between the theory research method the experimental result research method.

The theory research method: conclusion of the theories from related documents through sources such as e-book, lecture and instructor knowledge.

The experimental result research method: reality experience about the current graduate thesis process of Faculty for High Quality Training.

## Scientific and practical meaning of the project

The content of project has scientific meaning when research and conclusion of the current graduate thesis process of Faculty for High Quality Training. Thence, state the current process inadequacies and implement for computerize the current graduate thesis process to resolve those inadequacies.

Besides, the content of the project has practical meaning when applying the knowledge was researched and information technologies to build a graduate thesis management system on the web platform. Help to solve the inadequacies and make it more convenient for the current graduate thesis process.

# 

# CHAPTER 2. THEORETICAL BASIS

## The current graduate thesis process of Faculty of High Quality Training

### Participants and roles

The current graduate thesis process of Faculty of High Quality Training has three participant types, including:

* Head of Training: This participant type has the main responsibility and full permission during the graduate thesis work. It can decide the properties of the graduate thesis such as start time, end time, finish time of each state, participants, topics, etc. Sometimes, this participant type can work as a Lecturer participant type.
* Lecturer: This participant type has many roles in the graduate thesis, including:
  + Instructor: In this role, the lecturer can register one or more topics for the graduate thesis. When registered topics are accepted by the Head of training, the lecturer has the main responsibility in observing, following, and guide students to complete the topics which they choosen. Besides, this role also has permission to evaluate the result through the thesis implement process with account for 33% of the final topic result.
  + Reviewer: In this role, the lecturer can give a question for students who implement the topic. The scope of questions must be around the topic with the purpose survey the knowledge level of students about their topic. Head of Training can assign the Reviewer to review one topic which the Reviewer is not instructor to. In addition, the result of the reviewer can decide whether the topic should continue and account for 33% of the final topic result.
  + Defense council member: Generally, this role and the reviewer role quite similar to the evaluation method. The main difference is the member organization and evaluation method in the council (detail in the next section).
* Student: this participant type has a mission to complete the topic which was chosen by itself with the support of the topic instructor. Student can implement a topic with your partner who also participate in the graduate thesis or only alone.

### Phases

Normally, the graduate thesis process of Faculty of High Quality Training has 6 phases, including:

* Phase 1: Register topic for the lecturer.
* Phase 2: Register topic for the student.
* Phase 3: Semi-progress report.
* Phase 4: Review.
* Phase 5: Defense.
* Phase 6: Publish result.

#### Phase 1: Register topic for the lecturer

This is the first phase of the graduate thesis process and the related participant types are the Head of Training and the Lecturer. The main purpose of this phase is wants lecturers to create topics and to be an instructor for the graduate thesis. Sometimes, the topics can be continued development from topics that were implemented in Specialized essay.

In this phase, the Head of Training chooses and sends the invitation to lecturers who can give topics. After that, lecturers send topics to the Head of Training to get approval. If the Head of Training approves, topics will continue in the next phase. Otherwise, the Head of Training can send back and request the topic owner modify some information. In case the topic not suitable with criteria which are set by the Head of Training, the topic will be rejected.

#### Phase 2: Register topic for the student

The next phase of the graduate thesis process focuses on student participant type and lecturer participant type. The main purpose of this phase wants students can choose the topic which they will implement during the graduate thesis is processing. They can implement with their team, normally has two members, or implement only alone.

In this phase, students can contact the lecturer who is owner of the topic they want to choose for registration. If the lecturer accepts their registration, they can implement that topic and that topic can't accept any other registration. Opposite, if the lecturer rejects their registration and they still time to register, they can choose another topic for registration. Besides, the students don't need to register the topic which they implemented in the Specialized essay and continues development in the graduate thesis.

#### Phase 3: Semi-progress report

When the graduate thesis passed half the time, students must have a report for the Head of Training. The main purpose of this phase is the Head of Training wants to know what has done by students with their topic. Normally, the time and place of the reporting session are arranged by the Head of Training base on reality conditions and the progress of a graduate thesis.

In reporting, the students must prepare a report document for submitting and a presentation file for presentation. Content of the report document must have sections such as research scope, the amount of work done and difficulties encountered. Besides, the report document must have the signature of the instructor. In addition, the presentation file also has content similar to the report document but it's a slightly different presentation which depends on the student and should not exceed 15 minutes.

#### Phase 4: Review

When students passed in the semi-progress phase, this phase is the next phase in the graduate thesis. The main purpose of this phase wants to pre-check the complete product which was implemented by the student during the graduate thesis happening. Each topic will have time, place and one reviewer which is arranged by the Head of Training.

Like a mini defense, students must prepare a complete thesis document and a presentation file for presentation. During the presentation, the reviewer can give questions about the topic to students. Besides, with a target to contribute the topic more complete before the defense, the reviewer can raise ideas or comments to help students complete their product and thesis document.

The reviewer can evaluate the result of topic. The result of reviewer has criteria similar to member of defense council (detail at table 2.1 in the next section). After the review ended, the reviewer must submit the review document to the Head of Training to complete their mission

#### Phase 5: Defense

This phase is the most important phase in the graduate thesis process because it determines the thesis's result. Students will presentation to the defense council which is arranged by the Head of Training.

Members of the defense council also are arranged by the Head of Training and publish before the defense phase one or two weeks. Normally, the organizational composition of a defense council includes a chairman, a topic's reviewer and a commissioner.

Normally, the defense phase has activities like the review phase. But has a little bit different about time, each topic will have 10 minutes for presentation, 15 minutes for the demo and 10 minutes for question and answer about that topic.

At the end of the defense day, students can know the result from the defense council. The result is calculated according to the formula:

Each member of the defense council must evaluate the topic following 7 criteria which according to the table below

Table 2.1. Table evaluate criteria of the defense council

|  |  |  |
| --- | --- | --- |
| No | Criteria | Point ladder |
| 1 | The practicality of the topic, the understanding of the research problem | 10 |
| 2 | The correctness and reasonableness of the research method, of the design, of the solution is stated in the thesis. The degree of perfection of the product, the degree of student work completion | 40 |
| 3 | Quality of the presentation | 10 |
| 4 | Ability to read foreign language books for reference | 5 |
| 5 | Ability to synthesize knowledge, write a thesis | 10 |
| 6 | Quality in the form of the thesis (structure, format, spelling, ...) | 5 |
| 7 | Quality in response to the defense council’s questions | 20 |

Students can ask questions about the result after that if they not accept that result. After that, students must submit the thesis document to their school's library. If the thesis document has problems from the defense council, students will fix them before submission.

#### Phase 6: Publish result

In this phase, the individuals involved (include topics instructor, topics reviewer and topics defense council) will evaluate topics and submit results to the Training department. After that, students can view results on their student profile website.

The graduate thesis result of a student includes three parts, including topic's instructor part (1), topic's reviewer part (2), topic's defense council part (3) and is calculated according to the formula

If the graduate thesis result is greater than 5 points, the student is considered to pass. Opposite, student is considered to fail and will continue participate in the next graduate thesis.

## Overview of MySQL

### Introduction [1]

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

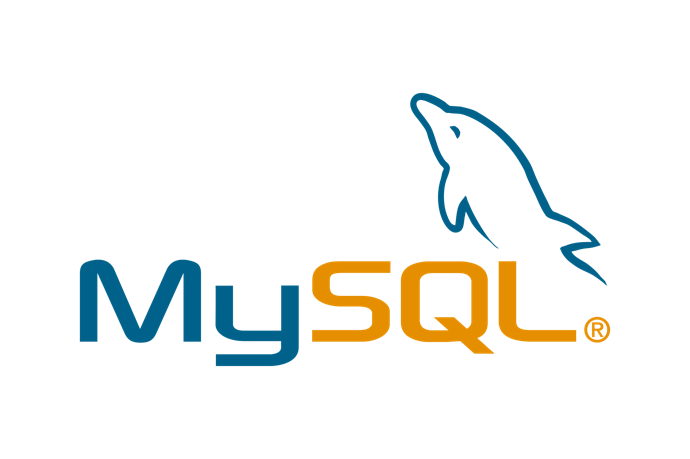


Figure 2.2.1. Logo of MySQL

### Features [1]

MySQL has following main features:

* Internals and Portability.
* Support many data types.
* Statements and Functions.
* Security.
* Scalability and Limits
* Connectivity.
* Localization.
* Clients and Tools

### History [1]

Started out with the intention of using the mSQL database system to connect to our tables using our own fast low-level (ISAM) routines. But after some testing, mSQL was not fast enough or flexible enough for needs. This resulted in a new SQL interface to our database but with almost the same API interface as mSQL. This API was designed to enable third-party code that was written for use with mSQL to be ported easily for use with MySQL.

MySQL is named after co-founder Monty Widenius's daughter, My.

The name of the MySQL Dolphin is “Sakila” which was chosen from a huge list of names suggested by users in our “Name the Dolphin” contest.

### Advantages [2]

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is becoming so popular because of many good reasons:

* MySQL is released under an open-source license. We have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

### MySQL in project

This project use MySQL 5.7 (provide by JawsDB) for the main database server which storage all data of application.

## Overview of Redis

### Introduction [3]

Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker. Redis has built-in replication, Lua scripting, LRU eviction, transactions and different levels of on-disk persistence, and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster.



Figure 2.2. Logo of Redis

### Features [3]

Redis has following main features:

* Supports data structures
* Transactions
* Pub/Sub
* Lua scripting
* Keys with a limited time-to-live
* LRU eviction of keys
* Automatic failover.

### History [4]

The name Redis means REmote DIctionary Server. The Redis project began when Salvatore Sanfilippo, nicknamed antirez, the original developer of Redis, was trying to improve the scalability of his Italian startup, developing a real-time web log analyzer. After encountering significant problems in scaling some types of workloads using traditional database systems, Sanfilippo began to prototype a first proof of concept version of Redis in Tcl. Later Sanfilippo translated that prototype to the C language and implemented the first data type, the list. After a few weeks of using the project internally with success, Sanfilippo decided to open source it, announcing the project on Hacker News. The project began to get traction, more so among the Ruby community, with GitHub and Instagram being among the first companies adopting it.

Sanfilippo was hired by VMware in March, 2010.

In May, 2013, Redis was sponsored by Pivotal Software (a VMware spin-off).

In June 2015, development became sponsored by Redis Labs.

In October 2018 Redis 5.0 was released, introducing Redis Stream - a new data structure that allows storage of multiple fields and string values with an automatic, time-based sequence at a single key.

In June 2020 Salvatore Sanfilippo stepped down as Redis maintainer.

### Advantages [5]

Following are certain advantages of Redis.

* Exceptionally fast − Redis is very fast and can perform about 110000 SETs per second, about 81000 GETs per second.
* Supports rich data types − Redis natively supports most of the datatypes that developers already know such as list, set, sorted set, and hashes. This makes it easy to solve a variety of problems as we know which problem can be handled better by which data type.
* Operations are atomic − All Redis operations are atomic, which ensures that if two clients concurrently access, Redis server will receive the updated value.
* Multi-utility tool − Redis is a multi-utility tool and can be used in a number of use cases such as caching, messaging-queues (Redis natively supports Publish/Subscribe), any short-lived data in your application, such as web application sessions, web page hit counts, etc.

### Redis in project

This project use Redis Cloud 6.0.5 (provide by redislabs) as a cache server. Support to storage SQL queries from client and return the cached result for faster performance of application.

## Overview of Object-relational mapping (ORM) and TypeORM

### Object-relational mapping (ORM) [6]

Object-relational mapping (ORM, O/RM, and O/R mapping tool) in computer science is a programming technique for converting data between incompatible type systems using object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language. There are both free and commercial packages available that perform object-relational mapping, although some programmers opt to construct their own ORM tools.

### TypeORM [7]

TypeORM is an ORM that can run in NodeJS, Browser, Cordova, PhoneGap, Ionic, React Native, NativeScript, Expo, and Electron platforms and can be used with TypeScript and JavaScript (ES5, ES6, ES7, ES8). Its goal is to always support the latest JavaScript features and provide additional features that help you to develop any kind of application that uses databases - from small applications with a few tables to large scale enterprise applications with multiple databases.

TypeORM supports both Active Record and Data Mapper patterns, unlike all other JavaScript ORMs currently in existence, which means you can write high quality, loosely coupled, scalable, maintainable applications the most productive way.

TypeORM is highly influenced by other ORMs, such as Hibernate, Doctrine and Entity Framework.

A picture containing text, tableware, dishware

Description automatically generated

Figure 2.2. Logo of TypeORM

### TypeORM in project

In this project, TypeORM is used to mapping data from MySQL to object type which compatible with Javascript – main programming language of project.

## Overview of NodeJS

### Introduction [8]

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

A picture containing text, sign, clock, clipart

Description automatically generated

Figure 2.3. Logo of NodeJS

### Features [8]

NodeJS has following main features:

* Asynchronous and Event Driven: All APIs of Node.js library is asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
* Very Fast: Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
* Single Threaded but Highly Scalable: Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
* No Buffering: Node.js applications never buffer any data. These applications simply output the data in chunks.
* License: Node.js is released under the MIT license.

### History [9]

Node.js was written initially by Ryan Dahl in 2009, about thirteen years after the introduction of the first server-side JavaScript environment, Netscape's LiveWire Pro Web. The initial release supported only Linux and Mac OS X. Its development and maintenance was led by Dahl and later sponsored by Joyent.

Dahl criticized the limited possibilities of the most popular web server in 2009, Apache HTTP Server, to handle a lot of concurrent connections (up to 10,000 and more) and the most common way of creating code (sequential programming), when code either blocked the entire process or implied multiple execution stacks in the case of simultaneous connections.

Dahl demonstrated the project at the inaugural European JSConf on 8 November 2009.[29][30][31] Node.js combined Google's V8 JavaScript engine, an event loop, and a low-level I/O API.

In January 2010, a package manager was introduced for the Node.js environment called npm. The package manager makes it easier for programmers to publish and share source code of Node.js packages and is designed to simplify installation, updating, and uninstallation of packages.

In June 2011, Microsoft and Joyent implemented a native Windows version of Node.js. The first Node.js build supporting Windows was released in July 2011.

In January 2012, Dahl stepped aside, promoting coworker and npm creator Isaac Schlueter to manage the project. In January 2014, Schlueter announced that Timothy J. Fontaine would lead the project.

In December 2014, Fedor Indutny started io.js, a fork of Node.js. Due to the internal conflict over Joyent's governance, io.js was created as an open governance alternative with a separate technical committee. Unlike Node.js, the authors planned to keep io.js up-to-date with the latest releases of the Google V8 JavaScript engine.

In February 2015, the intent to form a neutral Node.js Foundation was announced. By June 2015, the Node.js and io.js communities voted to work together under the Node.js Foundation.

In September 2015, Node.js v0.12 and io.js v3.3 were merged back together into Node v4.0. This merge brought V8 ES6 features into Node.js and a long-term support release cycle. As of 2016, the io.js website recommends that developers switch back to Node.js and that no further releases of io.js are planned due to the merge.

In 2019, the JS Foundation and Node.js Foundation merged to form the OpenJS Foundation.

### NodeJS in project

Because of the scalable nature of the project, NodeJS was chosen main platform. It was used in API server and graduate thesis application.

## Overview of TypeScript

### Introduction [10]

By definition, “TypeScript is JavaScript for application-scale development.”

TypeScript is a strongly typed, object oriented, compiled language. It was designed by Anders Hejlsberg (designer of C#) at Microsoft. TypeScript is both a language and a set of tools. TypeScript is a typed superset of JavaScript compiled to JavaScript. In other words, TypeScript is JavaScript plus some additional features.

A picture containing text, clipart

Description automatically generated

Figure 2.4. Logo of TypeScript

### Features [10]

TypeScript has following main features:

* TypeScript is just JavaScript. TypeScript starts with JavaScript and ends with JavaScript. Typescript adopts the basic building blocks of your program from JavaScript. Hence, you only need to know JavaScript to use TypeScript. All TypeScript code is converted into its JavaScript equivalent for the purpose of execution.
* TypeScript supports other JS libraries. Compiled TypeScript can be consumed from any JavaScript code. TypeScript-generated JavaScript can reuse all of the existing JavaScript frameworks, tools, and libraries.
* JavaScript is TypeScript. This means that any valid .js file can be renamed to .ts and compiled with other TypeScript files.
* TypeScript is portable. TypeScript is portable across browsers, devices, and operating systems. It can run on any environment that JavaScript runs on. Unlike its counterparts, TypeScript doesn’t need a dedicated VM or a specific runtime environment to execute.

### History [11]

TypeScript was first made public in October 2012 (at version 0.8), after two years of internal development at Microsoft. Soon after the announcement, Miguel de Icaza praised the language itself, but criticized the lack of mature IDE support apart from Microsoft Visual Studio, which was not available on Linux and OS X at that time. Today there is support in other IDEs, particularly in Eclipse, via a plug-in contributed by Palantir Technologies. Various text editors, including Emacs, Vim, Webstorm, Atom and Microsoft's own Visual Studio Code also support TypeScript.

TypeScript 0.9, released in 2013, added support for generics. TypeScript 1.0 was released at Microsoft's Build developer conference in 2014. Visual Studio 2013 Update 2 provides built-in support for TypeScript.

In July 2014, the development team announced a new TypeScript compiler, claiming 5× performance gains. Simultaneously, the source code, which was initially hosted on CodePlex, was moved to GitHub.

On 22 September 2016, TypeScript 2.0 was released; it introduced several features, including the ability for programmers to optionally prevent variables from being assigned null values, sometimes referred to as the billion-dollar mistake.

TypeScript 3.0 was released on 30 July 2018, bringing many language additions like tuples in rest parameters and spread expressions, rest parameters with tuple types, generic rest parameters and so on.

TypeScript 4.0 was released on 20th, August 2020. While 4.0 did not introduce any breaking changes, it added language features such as Custom JSX Factories and Variadic Tuple Types.

### TypeScript in project

All of source in this project is programed by TypeScript and compiled to Javascript for product. TypeScript help programer decrease bugs which root cause is data type or data structure.

## Overview of NestJS

### Introduction [12]

Nest (NestJS) is a framework for building efficient, scalable Node.js server-side applications. It uses progressive JavaScript, is built with and fully supports TypeScript (yet still enables developers to code in pure JavaScript) and combines elements of OOP (Object Oriented Programming), FP (Functional Programming), and FRP (Functional Reactive Programming).

Under the hood, Nest makes use of robust HTTP Server frameworks like Express (the default) and optionally can be configured to use Fastify as well.

Nest provides a level of abstraction above these common Node.js frameworks (Express/Fastify), but also exposes their APIs directly to the developer. This gives developers the freedom to use the myriad of third-party modules which are available for the underlying platform.



Figure 2.5. Logo of NestJS

### Features [12]

NestJS has following main features:

* Extensible: Gives the true flexibility by allowing use of any other libraries thanks to modular architecture.
* Versatile: An adaptable ecosystem that is a fully-fledged backbone for all kinds of server-side applications.
* Progressive: Takes advantage of latest JavaScript features, bringing design patterns and mature solutions to node.js world.

### NestJS in project

This project use NestJS framework to build API server. Therefore, it can be scalable in the future.

## Overview of React

### Introduction [13]

React is a library for building composable user interfaces. It encourages the creation of reusable UI components, which present data that changes over time. Lots of people use React as the V in MVC. React abstracts away the DOM from you, offering a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using React Native. React implements one-way reactive data flow, which reduces the boilerplate and is easier to reason about than traditional data binding.

Icon

Description automatically generated

Figure 2.6. Logo of React

### Features [13]

React has following main features:

* JSX − JSX is JavaScript syntax extension. It isn't necessary to use JSX in React development, but it is recommended. Besides, React can apply TypeScript with file extension TSX.
* Components − React is all about components. You need to think of everything as a component. This will help you maintain the code when working on larger scale projects.
* Unidirectional data flow and Flux − React implements one-way data flow which makes it easy to reason about your app. Flux is a pattern that helps keeping your data unidirectional.
* License − React is licensed under the Facebook Inc. Documentation is licensed under CC BY 4.0.

### Advantages [13]

* Uses virtual DOM which is a JavaScript object. This will improve apps performance, since JavaScript virtual DOM is faster than the regular DOM.
* Can be used on client and server side as well as with other frameworks.
* Component and data patterns improve readability, which helps to maintain larger apps.

### History [14]

React was created by Jordan Walke, a software engineer at Facebook, who released an early prototype of React called "FaxJS". He was influenced by XHP, an HTML component library for PHP. It was first deployed on Facebook's News Feed in 2011 and later on Instagram in 2012. It was open source at JSConf US in May 2013.

React Native, which enables native Android, iOS, and UWP development with React, was announced at Facebook's React Conf in February 2015 and open-sourced in March 2015.

On April 18, 2017, Facebook announced React Fiber, a new core algorithm of React library for building user interfaces. React Fiber was to become the foundation of any future improvements and feature development of the React library.

On September 26, 2017, React 16.0 was released to the public.

On February 16, 2019, React 16.8 was released to the public. The release introduced React Hooks.

On August 10, 2020, the React team announced the first release candidate for React v17.0, notable as the first major release without major changes to the React developer-facing API.

### React in project

This project using React (includes React hooks) to build the front-end of the graduate thesis application. Therefore, the application becomes easy to maintain and develop for developers. Besides, React helps end-user experiences better by dynamically rewriting the current web page with new data from the webserver, instead of the default method of the browser loading entire new pages.

## Overview of NextJS

### Introduction [15]

NextJS is the React framework. It helps the developer solve some problems which often occur when using React to build a web application, including:

Code has to be bundled using a bundler like webpack and transformed using a compiler like Babel.

production needs optimizations such as code splitting.

Statically pre-render some pages for performance and SEO and use server-side rendering or client-side rendering.

Write some server-side code to connect your React app to your data store.

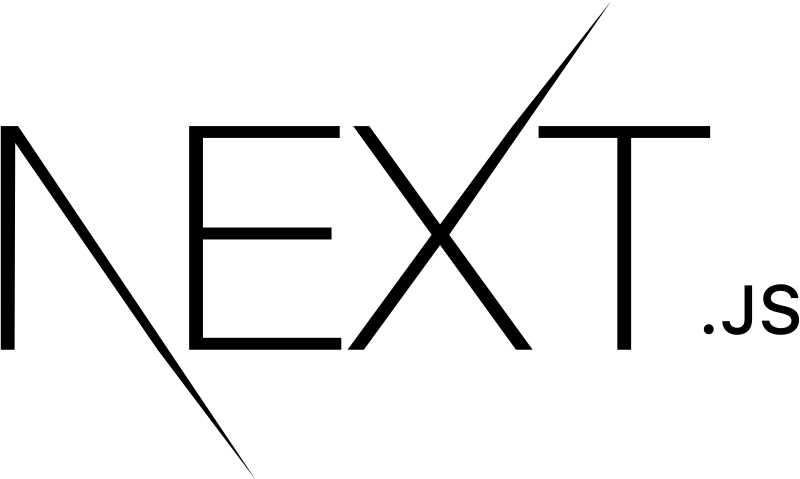


Figure 2.7. Logo of NextJS

### Features [16]

NextJS has following main features:

* Hot Code Reloading: Next.js reloads the page when it detects any change saved to disk.
* Automatic Routing: Any URL is mapped to the filesystem, to files put in the pages folder, and you don't need any configuration (you have customization options of course).
* Single File Components: Using styled-jsx, completely integrated as built by the same team, it's trivial to add styles scoped to the component.
* Server Rendering: We can render React components on the server-side, before sending the HTML to the client.
* Ecosystem Compatibility: Next.js plays well with the rest of the JavaScript, Node, and React ecosystem.
* Automatic Code Splitting: Pages are rendered with just the libraries and JavaScript that they need, no more. Instead of generating one single JavaScript file containing all the app code, the app is broken up automatically by Next.js in several different resources.
* Prefetching: The Link component, used to link together different pages, supports a prefetch prop which automatically prefetches page resources (including code missing due to code splitting) in the background.
* Dynamic Components: We can import JavaScript modules and React Components dynamically.
* Static Exports: Using the next export command, Next.js allows you to export a fully static site from your app.
* TypeScript Support: Next.js is written in TypeScript and as such comes with excellent TypeScript support.

### NextJS in project

This project uses NextJS combined with React to build the graduation thesis application. With NextJS features, this application easier to maintain and develop in the future. Besides, its assurance the performance of this application with SSG and SSR mechanism.

## Overview of Ant Design (for React)

### Introduction [17]

Ant Design is an enterprise-class UI design language and React UI library. It contains a set of high quality components and demos for building rich, interactive user interfaces.

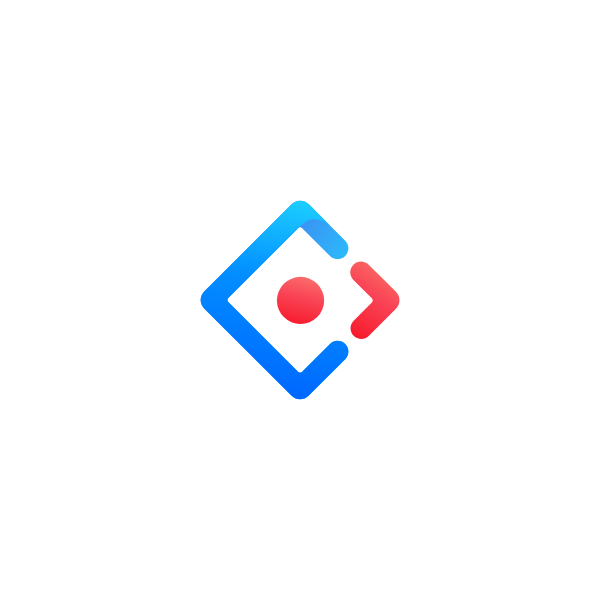


Figure 2.8. Logo of Ant Design

### Features [17]

Ant Design has following main features:

* Enterprise-class UI designed for web applications.
* A set of high-quality React components out of the box.
* Written in TypeScript with predictable static types.
* The whole package of design resources and development tools.
* Internationalization support for dozens of languages.
* Powerful theme customization in every detail.

### Ant Design in project

The graduate thesis application use Ant Design to main UI design.

## Other technologies and libraries

### Docker Engine [18]

Docker is a set of the platform as a service (PaaS) product that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels. All containers are run by a single operating system kernel and therefore use fewer resources than virtual machines.

The service has both free and premium tiers. The software that hosts the containers is called Docker Engine. It was first started in 2013 and is developed by Docker, Inc.

In this project, the author uses Docker to create a developed environment such as MySQL database and Redis.



Figure 2.9. Logo of Docker

### Continues Integration (CI) [19] and GitHub Actions [20]

Continuous Integration (CI) is a development practice where developers integrate code into a shared repository frequently, preferably several times a day. Each integration can then be verified by an automated build and automated tests. While automated testing is not strictly part of CI it is typically implied.

One of the key benefits of integrating regularly is that we can detect errors quickly and locate them more easily. As each change introduced is typically small, pinpointing the specific change that introduced a defect can be done quickly.

In recent years CI has become a best practice for software development and is guided by a set of key principles. Among them is a revision control, build automation and automated testing.

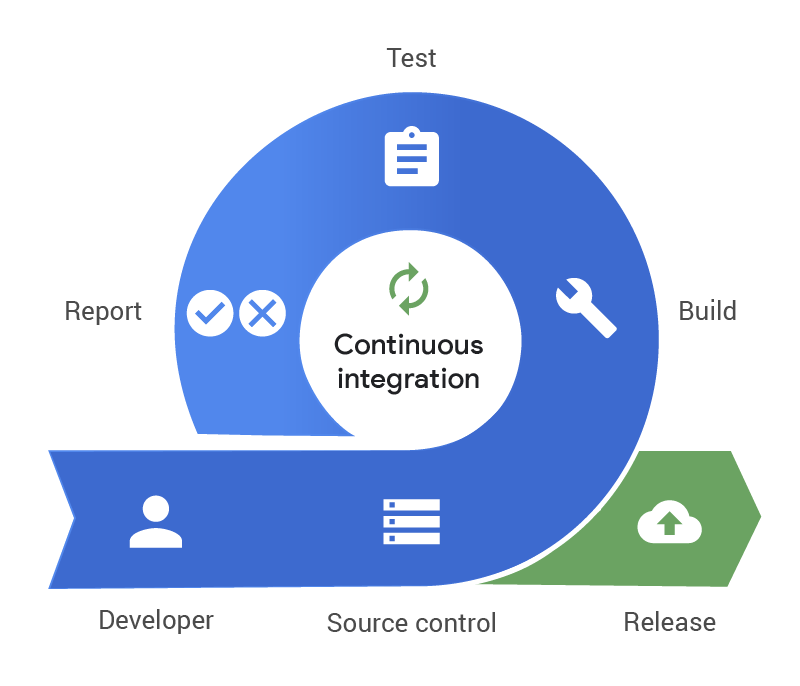


Figure 2.10. Continuous Integration life cycle

GitHub Actions is a feature of GitHub and be integrated in our repository on GitHub. We can automate, customize, and execute our software development workflows right in our repository with GitHub Actions. We can discover, create, and share actions to perform any job you'd like, including CI/CD, and combine actions in a completely customized workflow.

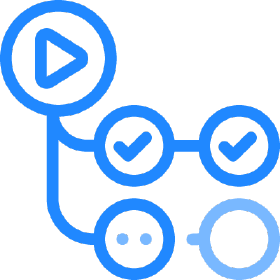


Figure 2.11. Logo of GitHub Actions

This project use GitHub Actions to checking two jobs:

* Lint job for assurance of the source convention.
* Build job for assurance new change of source can build to a product without bugs or problems.

### Amazon S3 [21]

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely tuned access controls to meet your specific business, organizational, and compliance requirements. Amazon S3 is designed for 99.999999999% (11 9's) of durability, and stores data for millions of applications for companies all around the world.

To assurance user's data, this project uses Amazon S3 to store files that are uploaded from users.

### Heroku [22]

Heroku is a platform as a service based on a managed container system, with integrated data services and a powerful ecosystem, for deploying and running modern apps. The Heroku developer experience is an app-centric approach for software delivery, integrated with today’s most popular developer tools and workflows.

Heroku runs your apps inside dynos — smart containers on a reliable, fully managed runtime environment. Developers deploy their code written in Node, Ruby, Java, PHP, Python, Go, Scala, or Clojure to a build system which produces an app that's ready for execution. The system and language stacks are monitored, patched, and upgraded, so it's always ready and up to date. The runtime keeps apps running without any manual intervention.

Heroku runs your app in lightweight, isolated Linux containers called "dynos." The platform offers different dyno types to help you get the best results for your type of app. With free version, heroku provided any services includes:

* 550-1,000 dyno hours per month.
* Deploy with Git and Docker.
* Custom domains.
* Container orchestration.
* Automatic OS patching.

This project uses Heroku as a main platform to deploy review and production including API server and the graduate thesis application. Besides, this project also uses add-ons of Heroku such as JawsDB MySQL (for database service) and Redis Cloud (for cache service) to support the operation.

Icon

Description automatically generated

Figure 2.12. Logo of Heroku

### Postman [23]

Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs faster.

Postman has main features:

* API Client: Quickly and easily send REST, SOAP, and GraphQL requests directly within Postman.
* Automated Testing: Automate manual tests and integrate them into your CI/CD pipeline to ensure that any code changes won't break the API in production.
* Design & Mock: Communicate the expected behavior of an API by simulating endpoints and their responses without having to set up a backend server.
* Documentation: Generate and publish beautiful, machine-readable documentation to make your API easier to consume.
* Monitors: Stay up to date on the health of your API by checking performance and response times at scheduled intervals.
* Workspaces: Provide a shared context for building and consuming APIs and collaborate in real-time with built-in version control.

This project uses Postman to develop and testing the API server.



Figure 2.13. Logo of Postman

# 

# CHAPTER 3. REQUIREMENT MODELING

## Define usecase

### Actor

Table 3.1. Actor

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Use case** | **Admin**  **Actor** | **Lecturer**  **Actor** | **Student**  **Actor** | **Guest**  **Actor** |
| 1 | Login |  |  |  | x |
| 2 | Logout | x | x | x |  |
| 3 | Change avatar | x | x | x |  |
| 4 | Manage lecturers | x |  |  |  |
| 5 | Manage students | x |  |  |  |
| 6 | Manage theses | x |  |  |  |
| 7 | Manage defense councils | x |  |  |  |
| 8 | Manage topics |  | x |  |  |
| 9 | Process topic approval request | x | x |  |  |
| 10 | Process topic register request |  | x | x |  |
| 11 | Comment in topic phase | x | x | x |  |
| 12 | Upload report file |  |  | x |  |
| 13 | Upload result file |  | x |  |  |
| 14 | View thesis result | x | x | x |  |
| 15 | Edit thesis result |  | x |  |  |
| 16 | Edit semi-progress report information | x |  |  |  |
| 17 | Edit semi-progress report result | x |  |  |  |
| 18 | Edit review information | x |  |  |  |
| 19 | Edit review result |  | x |  |  |
| 20 | Manage defense information | x |  |  |  |
| 21 | View lecturer detail | x | x | x |  |
| 22 | View student detail | x | x | x |  |
| 23 | View theses | x | x | x |  |
| 24 | View thesis | x | x | x |  |
| 25 | Search thesis | x | x | x |  |
| 26 | View topics | x | x | x |  |
| 27 | View topic | x | x | x |  |
| 28 | Search topic | x | x | x |  |

### Use case

Table 3.2. Use case

|  |  |  |  |
| --- | --- | --- | --- |
| No | Use case | | Use case ID |
| 1 | Login | | UC\_1 |
| 2 | Logout | | UC\_2 |
| 3 | Change avatar | | UC\_3 |
| 4 | Manage lecturers | | UC\_4 |
|  | 4.1 | Create lecturer | UC\_4.1 |
| 4.2 | Edit lecturer | UC\_4.2 |
| 4.3 | Delete lecturer | UC\_4.3 |
| 4.4 | Search lecturer | UC\_4.4 |
| 4.5 | View lecturers | UC\_4.5 |
| 5 | Manage students | | UC\_5 |
|  | 5.1 | Create student | UC\_5.1 |
| 5.2 | Edit student | UC\_5.2 |
| 5.3 | Delete student | UC\_5.3 |
| 5.4 | Search student | UC\_5.4 |
| 5.5 | View students | UC\_5.5 |
| 6 | Manage theses | | UC\_6 |
|  | 6.1 | Create thesis | UC\_6.1 |
| 6.2 | Edit thesis | UC\_6.2 |
| 6.3 | Delete thesis | UC\_6.3 |
| 6.4 | Active thesis | UC\_6.4 |
| 6.5 | Inactive thesis | UC\_6.5 |
| 7 | Manage defense councils | | UC\_7 |
|  | 7.1 | View defense councils | UC\_7.1 |
| 7.2 | Search defense council | UC\_7.2 |
| 7.3 | Create defense council | UC\_7.3 |
| 7.4 | Edit defense council | UC\_7.4 |
| 7.5 | Delete defense council | UC\_7.5 |
| 8 | Manage topics | | UC\_8 |
|  | 8.1 | Create topic | UC\_8.1 |
| 8.2 | Edit topic | UC\_8.2 |
| 8.3 | Delete topic | UC\_8.3 |
| 8.4 | Cancel topic | UC\_8.4 |
| 8.5 | Open register | UC\_8.5 |
| 8.4 | Close register | UC\_8.6 |
| 9 | Process topic approval request | | UC\_9 |
|  | 9.1 | Send approval request | UC\_9.1 |
| 9.2 | Withdraw approval request | UC\_9.2 |
| 9.3 | Accept approval request | UC\_9.3 |
| 9.4 | Reject approval request | UC\_9.4 |
| 9.5 | Sendback approval request | UC\_9.5 |
| 10 | Process topic register request | | UC\_10 |
|  | 10.1 | Send topic register request | UC\_10.1 |
| 10.2 | Accept topic register request | UC\_10.2 |
| 10.3 | Reject topic register request | UC\_10.3 |
| 11 | Comment in topic phase | | UC\_11 |
| 12 | Upload report file | | UC\_12 |
| 13 | Upload result file | | UC\_13 |
| 14 | View thesis result | | UC\_14 |
| 15 | Edit thesis result | | UC\_15 |
| 16 | Edit semi-progress report information | | UC\_16 |
| 17 | Edit semi-progress report result | | UC\_17 |
| 18 | Edit review information | | UC\_18 |
| 19 | Edit review result | | UC\_19 |
| 20 | Edit defense information | | UC\_20 |
| 21 | View lecturer detail | | UC\_21 |
| 22 | View student detail | | UC\_22 |
| 23 | View theses | | UC\_23 |
| 24 | View thesis | | UC\_24 |
| 25 | Search thesis | | UC\_25 |
| 26 | View topics | | UC\_26 |
| 27 | View topic | | UC\_27 |
| 28 | Search topic | | UC\_28 |

## Use case diagram

Figure 3.1. Use case diagram

## Use case specification

### Manage lecturers

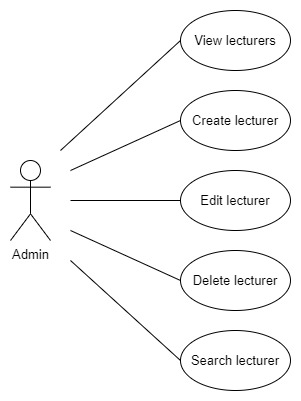


Figure 3.2. Use case manage lecturers

#### View lecturers

Table 3.3. Specific view lecturers

|  |  |
| --- | --- |
| Use Case ID | UC\_4.1 |
| Name | View lecturers |
| Goal | Allow admin views all lecturers in system |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the dashboard page, select “Quản lý giảng viên” |
| Exception | N/a |
| Open Issues | N/a |

#### Create lecturer

Table 3.4. Specific create lecturer

|  |  |
| --- | --- |
| Use Case ID | UC\_4.2 |
| Name | Create lecturer |
| Goal | Allow admin create a lecturer |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the “Quản lý giảng viên” page. 2. Click plus button. 3. Input new lecturer information. 4. Click “Xác nhận”. |
| Exception | - If require fields do not have value, click “Xác nhận” will occur error.  - If input value invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Edit lecturer

Table 3.5. Specific edit lecturer

|  |  |
| --- | --- |
| Use Case ID | UC\_4.3 |
| Name | Edit lecturer |
| Goal | Allow admin edit a lecturer |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the “Chi tiết giảng viên” page. 2. Click pen button. 3. Input new lecturer information 4. Click “Xác nhận” |
| Exception | - If require fields do not have value, click “Xác nhận” will throw error.  - If input value invalid, click “Xác nhận” will throw error. |
| Open Issues | N/a |

#### Delete lecturer

Table 3.6. Specific delete lecturer

|  |  |
| --- | --- |
| Use Case ID | UC\_4.4 |
| Name | Delete lecturer |
| Goal | Allow admin delete a lecturer |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the “Chi tiết giảng viên” page. 2. Click remove button. 3. Click “Xác nhận” |
| Exception | If lecturer does not exist, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Search lecturer

Table 3.7. Specific search lecturer

|  |  |
| --- | --- |
| Use Case ID | UC\_4.5 |
| Name | Search lecturer |
| Goal | Allow admin search one or many lecturers |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the “Quản lý giảng viên” page. 2. Input keyword into search bar. 3. Click glass button. |
| Exception | N/a |
| Open Issues | N/a |

### Manage students

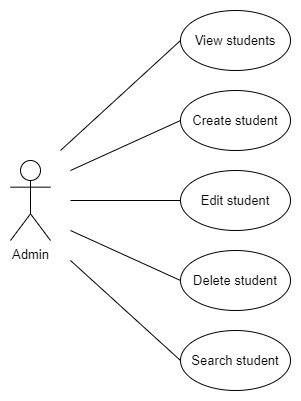


Figure 3.3. Use case manage students

#### View students

Table 3.8. Specific view students

|  |  |
| --- | --- |
| Use Case ID | UC\_5.1 |
| Name | View students |
| Goal | Allow admin views all students in system |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the dashboard page, select “Quản lý sinh viên” |
| Exception | N/a |
| Open Issues | N/a |

#### Create student

Table 3.9. Specific create student

|  |  |
| --- | --- |
| Use Case ID | UC\_5.2 |
| Name | Create student |
| Goal | Allow admin create a student |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the “Quản lý sinh viên” page. 2. Click plus button. 3. Input new student information. 4. Click “Xác nhận”. |
| Exception | - If require fields do not have value, click “Xác nhận” will occur error.  - If input value invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Edit student

Table 3.10. Specific edit student

|  |  |
| --- | --- |
| Use Case ID | UC\_5.3 |
| Name | Edit student |
| Goal | Allow admin edit a student |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the “Chi tiết sinh viên” page. 2. Click pen button. 3. Input new student information 4. Click “Xác nhận” |
| Exception | - If require fields do not have value, click “Xác nhận” will throw error.  - If input value invalid, click “Xác nhận” will throw error. |
| Open Issues | N/a |

#### Delete student

Table 3.11. Specific delete student

|  |  |
| --- | --- |
| Use Case ID | UC\_5.4 |
| Name | Delete student |
| Goal | Allow admin delete a student |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the “Chi tiết sinh viên” page. 2. Click remove button. 3. Click “Xác nhận” |
| Exception | If student does not exist, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Search student

Table 3.12. Specific search student

|  |  |
| --- | --- |
| Use Case ID | UC\_5.5 |
| Name | Search student |
| Goal | Allow admin search one or many students |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the “Quản lý sinh viên” page. 2. Input keyword into search bar. 3. Click glass button. |
| Exception | N/a |
| Open Issues | N/a |

### Manage theses

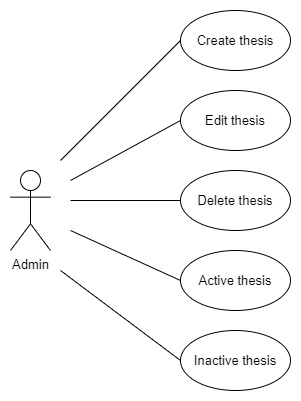


Figure 3.4. Use case manage theses

#### Create thesis

Table 3.13. Specific create thesis

|  |  |
| --- | --- |
| Use Case ID | UC\_6.1 |
| Name | Create thesis |
| Goal | Allow admin create a thesis |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the dashboard page, select “Khóa luận”. 2. Click plus button. 3. Input new thesis information. 4. Click “Xác nhận”. |
| Exception | - If require fields do not have value, click “Xác nhận” will occur error.  - If input value invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Edit thesis

Table 3.14. Specific edit thesis

|  |  |
| --- | --- |
| Use Case ID | UC\_6.2 |
| Name | Edit thesis |
| Goal | Allow admin edit a thesis |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the “Chi tiết khóa luận” page. 2. Click “Sửa khóa luận” button. 3. Input new thesis information 4. Click “Xác nhận” |
| Exception | - If require fields do not have value, click “Xác nhận” will throw error.  - If input value invalid, click “Xác nhận” will throw error.  - If target thesis has status is “Đang hoạt động”, click “Xác nhận” will throw error. |
| Open Issues | N/a |

#### Delete thesis

Table 3.15. Specific delete thesis

|  |  |
| --- | --- |
| Use Case ID | UC\_6.3 |
| Name | Delete thesis |
| Goal | Allow admin delete a thesis |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the “Chi tiết khóa luận” page. 2. Click “Xóa khóa luận” button. 3. Click “Xác nhận” |
| Exception | - If thesis does not exist, click “Xác nhận” will occur error.  - If target thesis has status is “Đang hoạt động”, click “Xác nhận” will throw error. |
| Open Issues | N/a |

#### Active thesis

Table 3.16. Specific active thesis

|  |  |
| --- | --- |
| Use Case ID | UC\_6.4 |
| Name | Active thesis |
| Goal | Allow admin active a thesis |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the “Chi tiết khóa luận” page. 2. Click “Kích hoạt” button. 3. Click “Xác nhận” button. |
| Exception | If thesis does not exist, click “Xác nhận” will throw error. |
| Open Issues | N/a |

#### Inactive thesis

Table 3.17. Specific inactive thesis

|  |  |
| --- | --- |
| Use Case ID | UC\_6.5 |
| Name | Inactive thesis |
| Goal | Allow admin inactive a thesis |
| Actors | Admin |
| Pre-conditions | Logged in as admin |
| Main flow | 1. On the “Chi tiết khóa luận” page. 2. Click “Ngưng kích hoạt” button. 3. Click “Xác nhận” button. |
| Exception | If thesis does not exist, click “Xác nhận” will throw error. |
| Open Issues | N/a |

### Manage defense councils

Diagram

Description automatically generated

Figure 3.5. Use case manage defense councils

#### View defense councils

Table 3.18. Specific view defense councils

|  |  |
| --- | --- |
| Use Case ID | UC\_7.1 |
| Name | View defense councils |
| Goal | Allow admin views all defense councils in system |
| Actors | Admin |
| Pre-conditions | - Logged in as admin.  - Current phase of target thesis is defense phase. |
| Main flow | 1. On the “Chi tiết khóa luận” page. 2. Click “Hội đồng bảo vệ” tab. |
| Exception | N/a |
| Open Issues | N/a |

#### Create defense council

Table 3.19. Specific create defense council

|  |  |
| --- | --- |
| Use Case ID | UC\_7.2 |
| Name | Create defense council |
| Goal | Allow admin create a defense council |
| Actors | Admin |
| Pre-conditions | - Logged in as admin.  - Current phase of target thesis is defense phase.  - Status of target thesis is active. |
| Main flow | 1. On the “Hội đồng bảo vệ” tab. 2. Click plus button. 3. Input new defense council information. 4. Click “Xác nhận”. |
| Exception | - If require fields do not have value, click “Xác nhận” will occur error.  - If input value invalid, click “Xác nhận” will occur error.  - If thesis phases invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Edit defense council

Table 3.20. Specific edit defense council

|  |  |
| --- | --- |
| Use Case ID | UC\_7.3 |
| Name | Edit student |
| Goal | Allow admin edit a defense council |
| Actors | Admin |
| Pre-conditions | - Logged in as admin.  - Current phase of target thesis is defense phase.  - Status of target thesis is active. |
| Main flow | 1. On the “Hội đồng bảo vệ” tab. 2. Click pen button. 3. Input new defense council information 4. Click “Xác nhận” |
| Exception | - If require fields do not have value, click “Xác nhận” will throw error.  - If input value invalid, click “Xác nhận” will throw error.  - If thesis phases invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Delete defense council

Table 3.21. Specific delete defense council

|  |  |
| --- | --- |
| Use Case ID | UC\_7.4 |
| Name | Delete student |
| Goal | Allow admin delete a defense council |
| Actors | Admin |
| Pre-conditions | - Logged in as admin.  - Current phase of target thesis is defense phase.  - Status of target thesis is active. |
| Main flow | 1. On the “Hội đồng bảo vệ” tab. 2. Click remove button. 3. Click “Xác nhận” |
| Exception | - If defense council does not exist, click “Xác nhận” will occur error.  - If thesis phases invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Search defense council

Table 3.22. Specific search defense council

|  |  |
| --- | --- |
| Use Case ID | UC\_7.5 |
| Name | Search defense council |
| Goal | Allow admin search one or many defense councils |
| Actors | Admin |
| Pre-conditions | Logged in as admin. |
| Main flow | 1. On the “Hội đồng bảo vệ” tab. 2. Input keyword into search bar. 3. Click glass button. |
| Exception | N/a |
| Open Issues | N/a |

### Manage topics

Diagram

Description automatically generated

Figure 3.6. Use case manage topics

#### Create topic

Table 3.23. Specific create topic

|  |  |
| --- | --- |
| Use Case ID | UC\_8.1 |
| Name | Create topic |
| Goal | Allow lecturer create a topic |
| Actors | Lecturer |
| Pre-conditions | - Logged in as lecturer.  - Be an attendee of target thesis.  - Current phase of target thesis is Topic register for lecturer.  - Status of target thesis is active. |
| Main flow | 1. On the “Danh sách đề tài” tab. 2. Click “Tạo đề tài” button. 3. Input new topic information. 4. Click “Xác nhận”. |
| Exception | - If require fields do not have value, click “Xác nhận” will occur error.  - If input value invalid, click “Xác nhận” will occur error.  - If thesis phases invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Edit topic

Table 3.24. Specific edit topic

|  |  |
| --- | --- |
| Use Case ID | UC\_8.2 |
| Name | Edit topic |
| Goal | Allow lecturer edit a topic |
| Actors | Lecturer |
| Pre-conditions | - Logged in as lecturer.  - Be an attendee of target thesis.  - Current phase of target thesis is Topic register for lecturer.  - State of target topic is “Mới” or “Thu hồi” or “Trả lại”.  - Status of target thesis is active. |
| Main flow | 1. On the “Danh sách đề tài” tab. 2. Click “Sửa đề tài” button. 3. Input new topic information. 4. Click “Xác nhận”. |
| Exception | If require fields do not have value, click “Xác nhận” will occur error.  - If input value invalid, click “Xác nhận” will occur error.  - If thesis phases invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Delete topic

Table 3.25. Specific delete topic

|  |  |
| --- | --- |
| Use Case ID | UC\_8.3 |
| Name | Delete topic |
| Goal | Allow lecturer delete a topic |
| Actors | Lecturer |
| Pre-conditions | - Logged in as lecturer.  - Be an attendee of target thesis.  - Current phase of target thesis is Topic register for lecturer.  - State of target topic is “Mới” or “Thu hồi” or “Trả lại”.  - Status of target thesis is active. |
| Main flow | 1. On the “Chi tiết đề tài” page. 2. Click “Xóa đề tài” button. 3. Click “Xác nhận” |
| Exception | If topic does not exist, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Cancel topic

Table 3.26. Specific cancel topic

|  |  |
| --- | --- |
| Use Case ID | UC\_8.4 |
| Name | Cancel topic |
| Goal | Allow lecturer cancel a topic |
| Actors | Lecturer |
| Pre-conditions | - Logged in as lecturer.  - Be an attendee of target thesis.  - Current phase of target thesis is Topic register for lecturer.  - State of target topic is “Mới” or “Thu hồi” or “Trả lại”.  - Status of target thesis is active. |
| Main flow | 1. On the “Chi tiết đề tài” page. 2. Click “Thông tin phê duyệt” tab. 3. Click “Hủy bỏ” button. 4. Click “Xác nhận” button. |
| Exception | - If topic does not exist, click “Xác nhận” will occur error.  - If status of topic invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Open register

Table 3.27. Specific open register

|  |  |
| --- | --- |
| Use Case ID | UC\_8.5 |
| Name | Open register |
| Goal | Allow lecturer open register topic |
| Actors | Lecturer |
| Pre-conditions | - Logged in as lecturer.  - Be an attendee of target thesis.  - Current phase of target thesis is Topic register for student.  - State of target topic is “Chấp nhận”.  - Status of target thesis is active. |
| Main flow | 1. On the “Chi tiết đề tài” page. 2. Click “Mở đăng ký” button. 3. Click “Xác nhận” button. |
| Exception | - If topic does not exist, click “Xác nhận” will occur error.  - If status of topic invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Close register

Table 3.28. Specific close register

|  |  |
| --- | --- |
| Use Case ID | UC\_8.6 |
| Name | Close register |
| Goal | Allow lecturer close register topic |
| Actors | Lecturer |
| Pre-conditions | - Logged in as lecturer.  - Be an attendee of target thesis.  - Current phase of target thesis is Topic register for student.  - State of target topic is “Chấp nhận”.  - Status of target thesis is active. |
| Main flow | 1. On the “Chi tiết đề tài” page. 2. Click “Đóng đăng ký” button. 3. Click “Xác nhận” button. |
| Exception | - If topic does not exist, click “Xác nhận” will occur error.  - If status of topic invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

### Process topic approval request

Diagram

Description automatically generated

Figure 3.7. Use case process topic approval request

#### Send approval request

Table 3.29. Specific send approval request

|  |  |
| --- | --- |
| Use Case ID | UC\_9.1 |
| Name | Send approval request |
| Goal | Allow lecturer send approval request to admin |
| Actors | Lecturer |
| Pre-conditions | - Logged in as lecturer.  - Be an attendee of target thesis.  - Current phase of target thesis is Topic register for lecturer.  - Status of target thesis is active.  - Target topic has owner by actor.  - State of target topic is “Mới” or “Thu hồi” or “Trả lại”. |
| Main flow | 1. On the “Thông tin phê duyệt” tab. 2. Click “Yêu cầu phê duyệt” button. 3. Click “Xác nhận”. |
| Exception | If thesis phases invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Withdraw approval request

Table 3.30. Specific withdraw approval request

|  |  |
| --- | --- |
| Use Case ID | UC\_9.2 |
| Name | Withdraw approval request |
| Goal | Allow lecturer withdraw approval request |
| Actors | Lecturer |
| Pre-conditions | - Logged in as lecturer.  - Be an attendee of target thesis.  - Current phase of target thesis is Topic register for lecturer.  - Status of target thesis is active.  - Target topic has owner by actor.  - State of target topic is “Đang được phê duyệt”. |
| Main flow | 1. On the “Thông tin phê duyệt” tab. 2. Click “Thu hồi” button. 3. Click “Xác nhận”. |
| Exception | If thesis phases invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Accept approval request

Table 3.31. Specific accept approval request

|  |  |
| --- | --- |
| Use Case ID | UC\_9.3 |
| Name | Accept approval request |
| Goal | Allow admin accept approval request of lecturer |
| Actors | Admin |
| Pre-conditions | - Logged in as admin.  - Current phase of target thesis is Topic register for lecturer.  - Status of target thesis is active.  - State of target topic is “Đang được phê duyệt”. |
| Main flow | 1. On the “Thông tin phê duyệt” tab. 2. Click “Chấp nhận” button. 3. Click “Xác nhận”. |
| Exception | If thesis phases invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Reject approval request

Table 3.32. Specific reject approval request

|  |  |
| --- | --- |
| Use Case ID | UC\_9.4 |
| Name | Reject approval request |
| Goal | Allow admin reject approval request of lecturer |
| Actors | Admin |
| Pre-conditions | - Logged in as admin.  - Current phase of target thesis is Topic register for lecturer.  - Status of target thesis is active.  - State of target topic is “Đang được phê duyệt”. |
| Main flow | 1. On the “Thông tin phê duyệt” tab. 2. Click “Từ chối” button. 3. Click “Xác nhận”. |
| Exception | If thesis phases invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Sendback approval request

Table 3.33. Specific sendback approval request

|  |  |
| --- | --- |
| Use Case ID | UC\_9.5 |
| Name | Sendback approval request |
| Goal | Allow admin sendback approval request of lecturer |
| Actors | Admin |
| Pre-conditions | - Logged in as admin.  - Current phase of target thesis is Topic register for lecturer.  - Status of target thesis is active.  - State of target topic is “Đang được phê duyệt”. |
| Main flow | 1. On the “Thông tin phê duyệt” tab. 2. Click “Trả lại” button. 3. Click “Xác nhận”. |
| Exception | If thesis phases invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

### Process topic register request

Diagram

Description automatically generated

Figure 3.8. Use case process topic register request

#### Send register request

Table 3.34. Specific send register request

|  |  |
| --- | --- |
| Use Case ID | UC\_10.1 |
| Name | Send register request |
| Goal | Allow student send register request to lecturer |
| Actors | Student |
| Pre-conditions | - Logged in as student.  - Be an attendee of target thesis.  - Current phase of target thesis is Topic register for student.  - Status of target thesis is active.  - Status of target topic is “Mở đăng ký”.  - Actor have not registered any topic. |
| Main flow | 1. On the “Danh sách sinh viên đăng ký đề tài” tab. 2. Click “Đăng ký đề tài” button. 3. Click “Xác nhận”. |
| Exception | If thesis phases invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Accept register request

Table 3.35. Specific accept register request

|  |  |
| --- | --- |
| Use Case ID | UC\_10.2 |
| Name | Accept register request |
| Goal | Allow lecturer accept register request of student |
| Actors | Lecturer |
| Pre-conditions | - Logged in as lecturer.  - Be an attendee of target thesis.  - Current phase of target thesis is Topic register for student.  - Status of target thesis is active.  - Status of target topic is “Mở đăng ký”.  - Target topic is owner by actor. |
| Main flow | 1. On the “Danh sách sinh viên đăng ký đề tài” tab. 2. Click accept icon button. 3. Click “Xác nhận”. |
| Exception | - If thesis phases invalid, click “Xác nhận” will occur error.  - If current registered student amount equal or greater than maximum student amount, click “Xác nhận” will occur error. |
| Open Issues | N/a |

#### Reject register request

Table 3.36. Specific reject register request

|  |  |
| --- | --- |
| Use Case ID | UC\_10.3 |
| Name | Reject register request |
| Goal | Allow lecturer reject register request of student |
| Actors | Lecturer |
| Pre-conditions | - Logged in as lecturer.  - Be an attendee of target thesis.  - Current phase of target thesis is Topic register for student.  - Status of target thesis is active.  - Status of target topic is “Mở đăng ký”.  - Target topic is owner by actor. |
| Main flow | 1. On the “Danh sách sinh viên đăng ký đề tài” tab. 2. Click remove icon button. 3. Click “Xác nhận”. |
| Exception | If thesis phases invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

### Edit review information



Figure 3.9. Use case edit review information

Table 3.37. Specific edit review information

|  |  |
| --- | --- |
| Use Case ID | UC\_18 |
| Name | Edit review information |
| Goal | Allow admin edit review information |
| Actors | Admin |
| Pre-conditions | - Logged in as admin.  - Current phase of target thesis is review phase.  - Status of target thesis is active. |
| Main flow | 1. On the “Phản biện” tab. 2. Click “Chỉnh sửa” button. 3. Input new review information. 4. Click “Xác nhận”. |
| Exception | - If thesis phases invalid, click “Xác nhận” will occur error.  - If input value invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

### Edit defense information

Diagram

Description automatically generated

Figure 3.10. Use case edit defense information

Table 3.38. Specific edit defense information

|  |  |
| --- | --- |
| Use Case ID | UC\_20 |
| Name | Edit review information |
| Goal | Allow admin edit defense information |
| Actors | Admin |
| Pre-conditions | - Logged in as admin.  - Current phase of target thesis is defense phase.  - Status of target thesis is active. |
| Main flow | 1. On the “Bảo vệ” tab. 2. Click “Chỉnh sửa” button. 3. Input new defense information. 4. Click “Xác nhận”. |
| Exception | - If thesis phases invalid, click “Xác nhận” will occur error.  - If input value invalid, click “Xác nhận” will occur error. |
| Open Issues | N/a |

# 

# CHAPTER 4. SOFTWARE DESIGN

## **System design**

### **Workflow**

Diagram

Description automatically generated

Figure 4.1. Workflow about graduate thesis

## **Database design**

### **Database diagram**

Reference to Appendix 1

### Description for each table

Table 4.1. Description for user table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage user information | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK |  | Unique key, auto increment |
| 2 | username | varchar(50) |  |  |  |
| 3 | firstname | varchar(50) |  | ✓ |  |
| 4 | lastname | varchar(50) |  | ✓ |  |
| 5 | gender | tinyint(4) |  | ✓ |  |
| 6 | email | varchar(100) |  | ✓ |  |
| 7 | address | varchar(100) |  | ✓ |  |
| 8 | phone | char(10) |  | ✓ |  |
| 9 | status | tinyint(4) |  |  | 1 – Inactive  2 – Active  Default: 2 |
| 10 | is\_admin | tinyint(4) |  |  | 1 – False  2 – True  Default: 1 |
| 11 | user\_type | tinyint(4) |  |  | 1 – Student  2 – Lecturer |
| 12 | password | varchar(40) |  |  | Hashed by SHA1 algorithm |
| 13 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 14 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 15 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.2. Description for user\_refresh\_token table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage refresh token of user | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK |  | Unique key, auto increment |
| 2 | userId | varchar(50) | FK |  | Map to id of user table |
| 3 | browser | varchar(50) |  | ✓ | Browser name |
| 4 | version | varchar(50) |  | ✓ | Browser version |
| 5 | platform | varchar(50) |  | ✓ | Platform of client |
| 6 | os | varchar(50) |  | ✓ | Operation system of client |
| 7 | refreshToken | varchar(255) |  | ✓ | Refresh token |
| 8 | source | varchar(255) |  | ✓ | User-agent header |
| 9 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 10 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 11 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.3. Description for lecturer table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage lecturer information | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK  FK |  | - Unique key, auto increment.  - Map to id of user table |
| 2 | lecturer\_id | varchar(4) |  | ✓ | ID of lecturer |
| 3 | position | varchar(255) |  | ✓ |  |
| 4 | level | varchar(255) |  | ✓ |  |
| 5 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 6 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 7 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.4. Description for student table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage student information | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK  FK |  | - Unique key, auto increment.  - Map to id of user table |
| 2 | student\_id | char(8) |  | ✓ | ID of student |
| 3 | school\_year | varchar(10) |  | ✓ |  |
| 4 | student\_class | varchar(20) |  | ✓ |  |
| 5 | is\_graduate | tinyint(4) |  |  | 1 – False  2 – True  Default: 1 |
| 6 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 7 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 8 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.5. Description for thesis table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage thesis information | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK  FK |  | -Unique key, auto increment.  - Map to id of user table |
| 2 | subject | varchar(100) |  |  |  |
| 3 | creator\_id | int(11) | FK |  | Map to id of lecturer table |
| 4 | start\_time | datetime |  |  |  |
| 5 | end\_time | datetime |  |  |  |
| 6 | state | tinyint(4) |  |  |  |
| 7 | lecturer\_topic\_register | datetime |  |  |  |
| 8 | student\_topic\_register | datetime |  |  |  |
| 9 | progress\_report | datetime |  |  |  |
| 10 | review | datetime |  |  |  |
| 11 | defense | datetime |  |  |  |
| 12 | status | tinyint(4) |  |  | 1 – Inactive  2 – Active  Default: 1 |
| 13 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 14 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 15 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.6. Description for thesis\_lecturer table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage lecturer type participant information | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | thesis\_id | int(11) | PK  FK |  | Map to id of thesis table |
| 2 | lecturer\_id | int(11) | PK  FK |  | Map to id of lecturer table |
| 3 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 4 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 5 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.7. Description for thesis\_student table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage student type participant information | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | thesis\_id | int(11) | PK  FK |  | Map to id of thesis table |
| 2 | student\_id | int(11) | PK  FK |  | Map to id of student table |
| 3 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 4 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 5 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.8. Description for topic table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage topic information | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK |  | Unique key, auto increment. |
| 2 | subject | text |  |  |  |
| 3 | creator\_id | int(11) | FK |  | Map to id of lecturer table |
| 4 | description | text |  |  |  |
| 5 | status | tinyint(4) |  |  |  |
| 6 | approver\_id | int(11) | FK |  | Map to id of lecturer table |
| 7 | thesis\_id | int(11) | FK |  | Map to id of thesis table |
| 8 | max\_student | tinyint(4) |  |  | Default: 2 |
| 9 | current\_student | tinyint(4) |  |  | Default: 0 |
| 10 | register\_status | tinyint(4) |  |  | 1 – Close  2 – Open  Default: 1 |
| 11 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 12 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 13 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.9. Description for topic\_state table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage state action of topic | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK |  | Unique key, auto increment. |
| 2 | topic\_id | int(11) | FK |  | Map to id of topic table |
| 3 | processor\_id | int(11) | FK |  | Map to id of lecturer table |
| 4 | note | text |  | ✓ |  |
| 5 | action | tinyint(4) |  |  | 1 – New  2 – Approved  3 – Rejected  4 – Sendback  5 – Withdraw  6 – Send request  7 – Canceled  Default: 1 |
| 6 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 7 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 8 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.10. Description for topic\_student table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage students who register topic | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK |  | Unique key, auto increment. |
| 2 | topic\_id | int(11) | FK |  | Map to id of topic table |
| 3 | student\_id | int(11) | FK |  | Map to id of student table |
| 4 | status | tinyint(4) |  |  | 1 – Peding  2 – Approved  3 – Rejected  Default: 1 |
| 5 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 6 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 7 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.11. Description for progress\_report table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage semi-progress report information | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK |  | Unique key, auto increment. |
| 2 | time | datetime |  |  |  |
| 3 | place | varchar(100) |  |  |  |
| 4 | result | tinyint(4) |  |  | 1 – Not decided  2 – Passed  3 – Failed  Default: 1 |
| 5 | note | text |  | ✓ |  |
| 6 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 7 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 8 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.12. Description for review table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage review information | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK |  | Unique key, auto increment. |
| 2 | time | datetime |  |  |  |
| 3 | place | varchar(100) |  |  |  |
| 4 | result | tinyint(4) |  |  | 1 – Not decided  2 – Passed  3 – Failed  Default: 1 |
| 5 | note | text |  | ✓ |  |
| 6 | reviewer\_id | int(11) | FK |  | Map to id of lecturer table |
| 7 | reviewer\_comment | text |  | ✓ |  |
| 8 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 9 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 10 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.13. Description for council table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage council information | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK |  | Unique key, auto increment. |
| 2 | name | varchar(100) |  |  |  |
| 3 | thesis\_id | int(11) | FK |  | Map to id of thesis table |
| 4 | chairman\_id | int(11) | FK |  | Map to id of lecturer table |
| 5 | instructor\_id | int(11) | FK |  | Map to id of lecturer table |
| 6 | commissioner\_id | int(11) | FK |  | Map to id of lecturer table |
| 8 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 9 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 10 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.14. Description for defense table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage defense information | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK |  | Unique key, auto increment. |
| 2 | time | datetime |  |  |  |
| 3 | place | varchar(100) |  |  |  |
| 4 | note | text |  | ✓ |  |
| 5 | council\_id | int(11) | FK |  | Map to id of council table |
| 6 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 7 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 8 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.15. Description for result table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage result information | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK |  | Unique key, auto increment. |
| 2 | topic\_id | int(11) | FK |  | Map to id of topic table |
| 3 | student\_id | int(11) | FK |  | Map to id of student table |
| 4 | creator\_id | int(11) | FK |  | Map to id of lecturer table |
| 5 | note | text |  | ✓ |  |
| 6 | type | tinyint(4) |  |  | 1 – Instructor  2 – Review  3 – Defense |
| 7 | point | json |  |  | {  title: string,  rate: number,  value: null | number  } |
| 8 | status | tinyint(4) |  |  | 1 – Lock  2 – Unlock  Default: 1 |
| 9 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 10 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 11 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.16. Description for comment table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage comment information | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK |  | Unique key, auto increment. |
| 2 | topic\_id | int(11) | FK |  | Map to id of topic table |
| 4 | creator\_id | int(11) | FK |  | Map to id of user table |
| 5 | content | text |  |  |  |
| 6 | mode | tinyint(4) |  |  | 1 – Private  2 – Public  Default: 2 |
| 7 | module | tinyint(4) |  |  | 1 – Semi-progress report  2 – Review  3 – Defense |
| 8 | deleted\_at | datetime(6) |  | ✓ | Delete datetime |
| 9 | created\_at | datetime(6) |  | ✓ | Create datetime |
| 10 | updated\_at | datetime(6) |  | ✓ | Updated datetime |

Table 4.17. Description for migrations table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purpose: Storage migrations log | | | | | |
| No | **Column** | **Type** | **Key** | **Allow Null** | **Description** |
| 1 | id | int(11) | PK |  | Unique key, auto increment. |
| 2 | timestamp | bigint(20) |  |  |  |
| 3 | name | varchar(255) |  |  |  |

## **Interface design**

### **Screen flow**

Reference to Appendix 2

### **Specification of the screens**

This section only describes the specification of the main screens of the graduation thesis application. Therefore, some screens describe in the screen flow of section 4.4.1 will be not described.

#### Thesis list screen

##### Thesis list screen

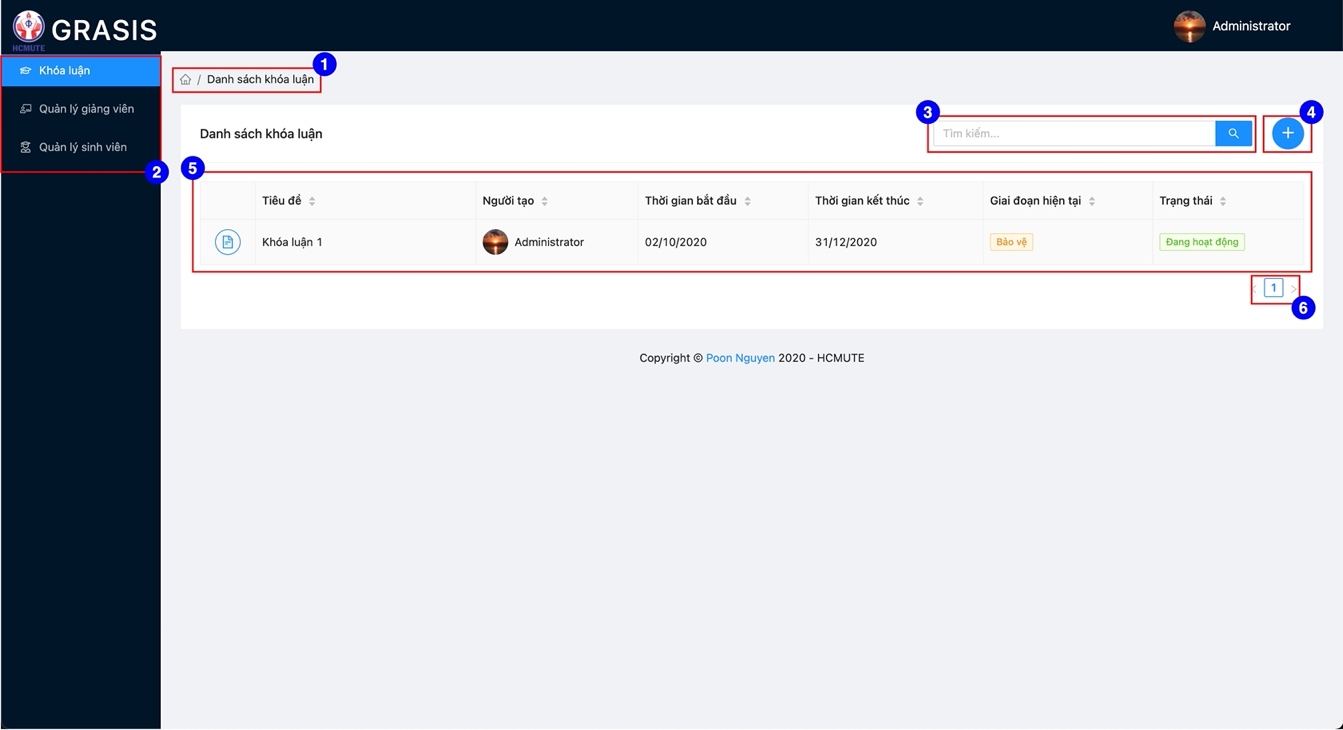


Figure 4.2. Thesis list screen

Table 4.18. Thesis list screen specification

|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Page breadcrumbs | When the user clicks on Item-1, the user will be redirected to corresponding page. |
| Item-2 | Navigation bar | When the user clicks on Item-2, the user will be redirected to corresponding page. |
| Item-3 | Search bar | When the user input keyword and clicks glass icon, search result will be displayed on Item-5. |
| Item-4 | Add thesis button | When the user clicks on Item-4, the user will be redirected to “Tạo khóa luận” page. |
| Item-5 | Pagination button | When the user clicks on Item-5, the thesis list will be displayed on Item-4 by page number. |

##### Create thesis screen

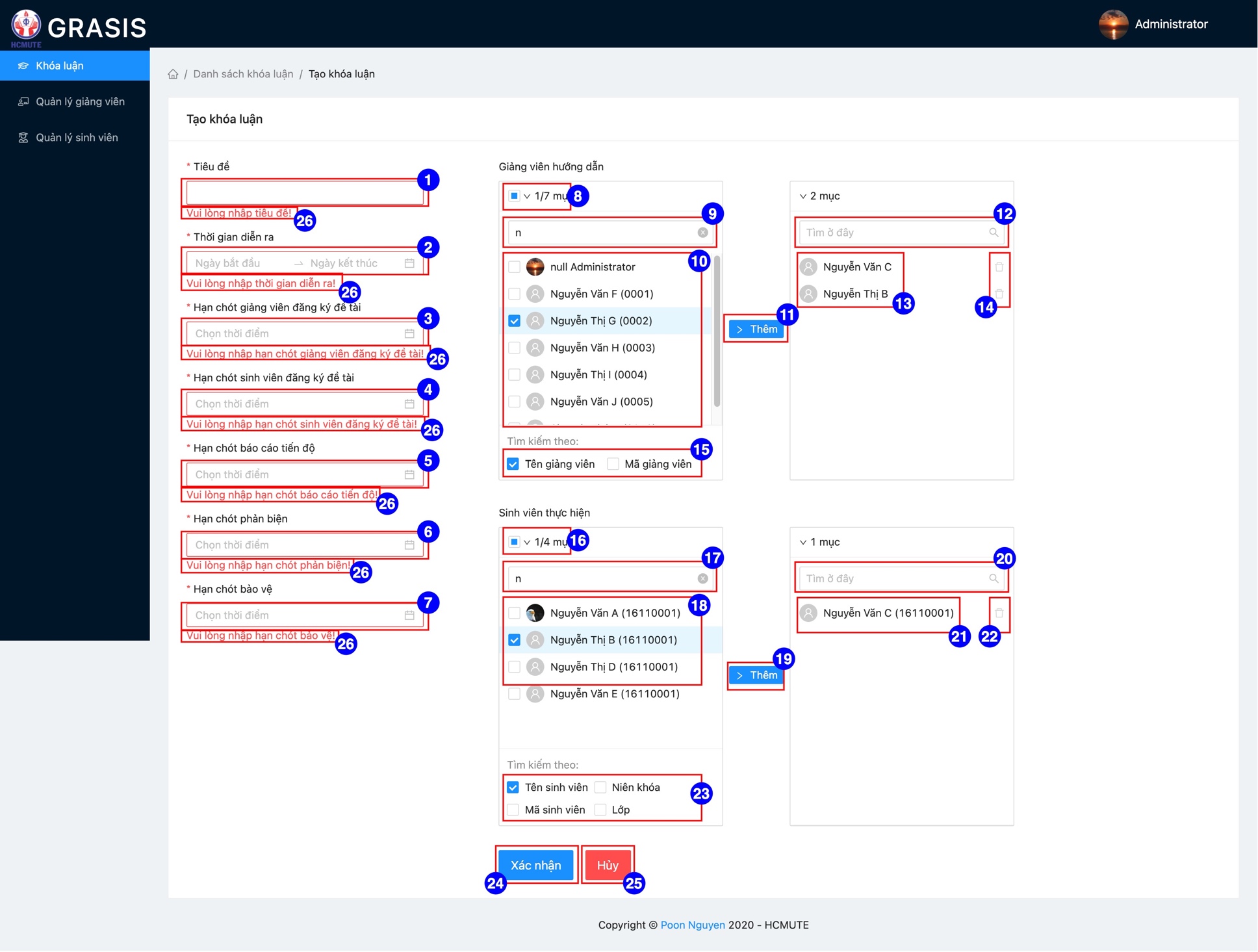


Figure 4.3. Create thesis screen

Table 4.19. Create thesis screen specification

|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Subject |  |
| Item-2 | Duration time | When the user clicks on Item-2, selecting data dialog will display. |
| Item-3 | Deadline of register topic for lecturer time | When the user clicks on Item-3, selecting data dialog will display. |
| Item-4 | Deadline of register topic for student time | When the user clicks on Item-4, selecting data dialog will display. |
| Item-5 | Deadline of semi-progress report time | When the user clicks on Item-5, selecting data dialog will display. |
| Item-6 | Deadline of review time | When the user clicks on Item-6, selecting data dialog will display. |
| Item-7 | Deadline of defense time | When the user clicks on Item-7, selecting data dialog will display. |
| Item-8 | Select all lecturers in Item-10 | When the user clicks on Item-8, all lecturersitems in Item-10 will be selected. |
| Item-9 | Suggestion lecturer search bar | When the user inputs keyword on Item-9, search result will display at Item-10 |
| Item-10 | Suggestion lecturer search result |  |
| Item-11 | Add lecturer button | When the user clicks on Item-11, selected lecturers at Item-10 will move to Item-13 |
| Item-12 | Selected lecturer search bar | When the user inputs keyword on Item-12, search result will display at Item-13 |
| Item-13 | Selected lecturers |  |
| Item-14 | Remove selected lecturer button | When the user clicks on Item-14, target lecture will move to Item-10 |
| Item-15 | Criteria for suggestion lecturer search |  |
| Item-16 | Select all students in Item-18 | When the user clicks on Item-16, all students in Item-10 will be selected. |
| Item-17 | Suggestion student search bar |  |
| Item-18 | Suggestion student search result |  |
| Item-19 | Add student button | When the user clicks on Item-19, selected students at Item-18 will move to Item-21 |
| Item-20 | Selected student search bar | When the user inputs keyword on Item-20, search result will display at Item-18 |
| Item-21 | Selected students |  |
| Item-22 | Remove selected student button | When the user clicks on Item-22, target student will move to Item-18 |
| Item-23 | Criteria for suggestion student search |  |
| Item-24 | Submit button | When the user clicks on Item-24, success notification will display and redirect to thesis list screen |
| Item-25 | Cancel button | When the user clicks on Item-24, redirect to thesis list screen |
| Item-26 | Validation error message | When the user inputs invalid or emptu value, Item-26 will display. |

#### Thesis detail screen

##### Thesis detail tab

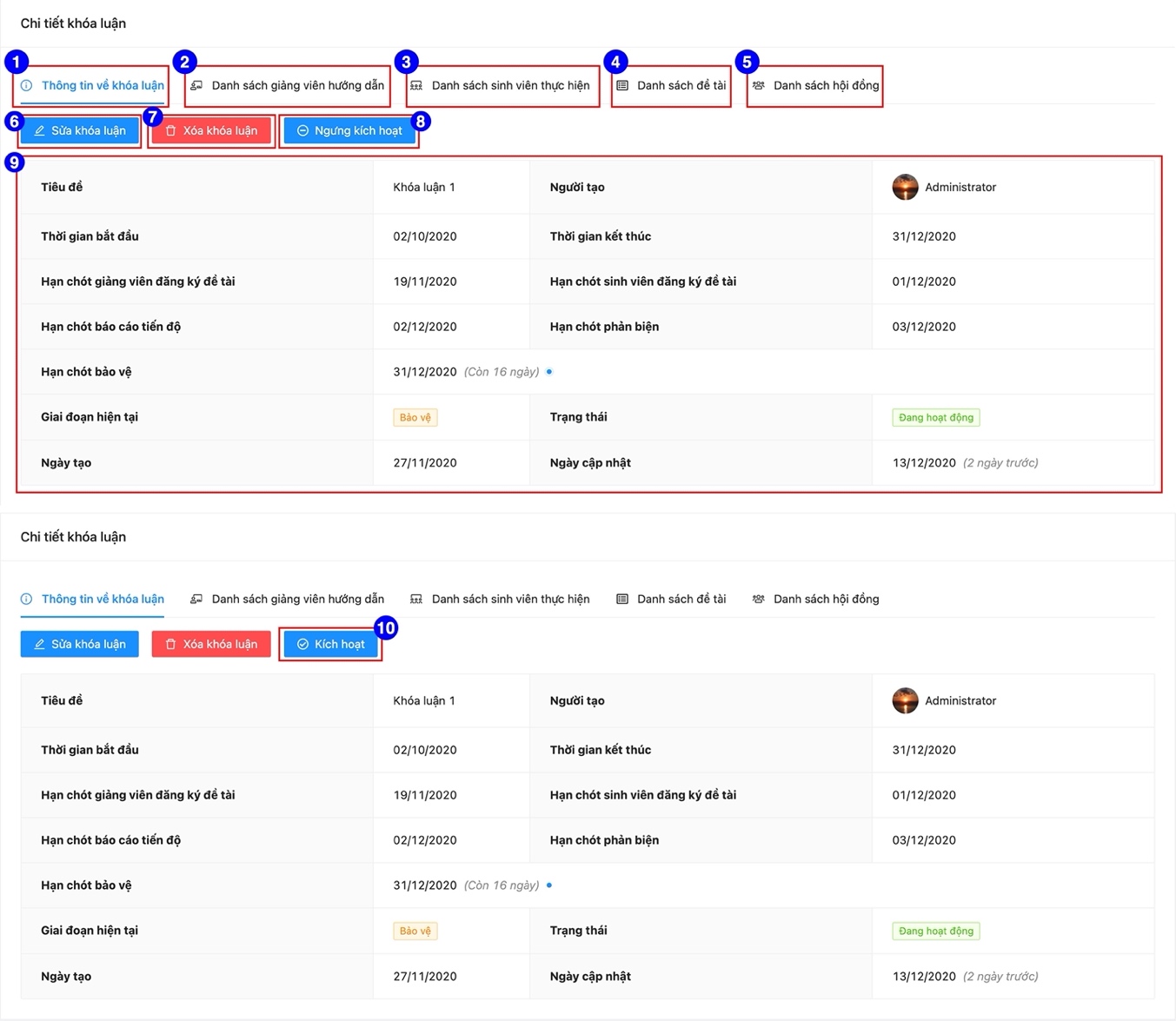


Figure 4.4. Thesis detail screen

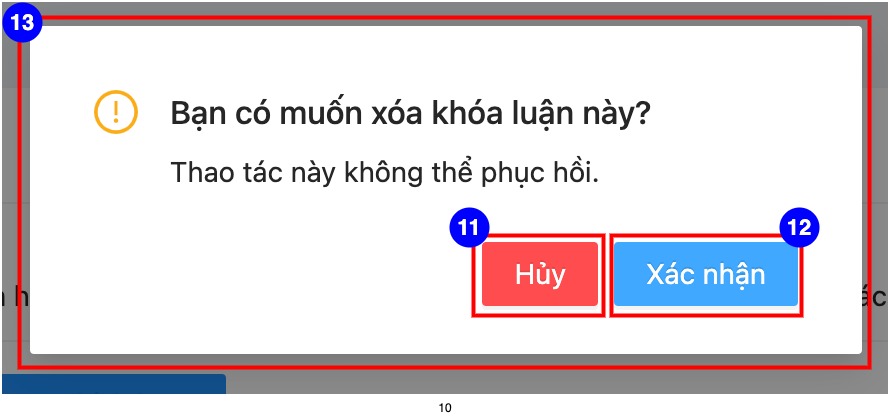


Figure 4.5. Delete thesis confirm dialog

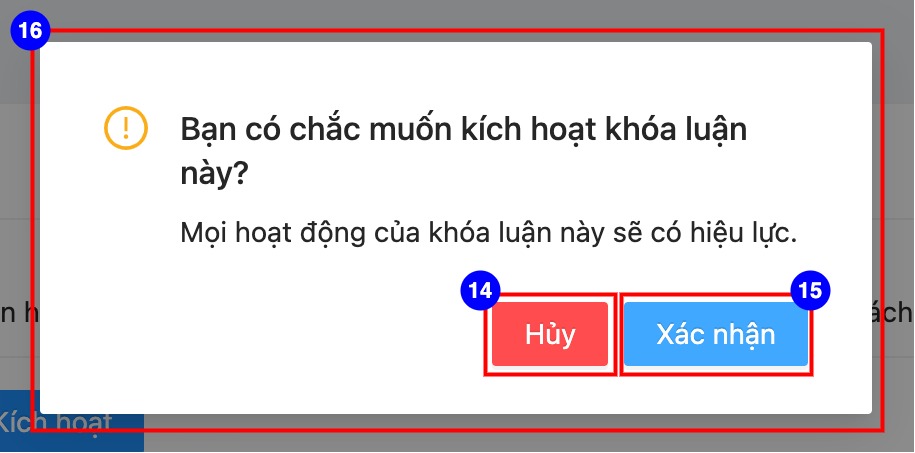


Figure 4.6. Active thesis confirm dialog

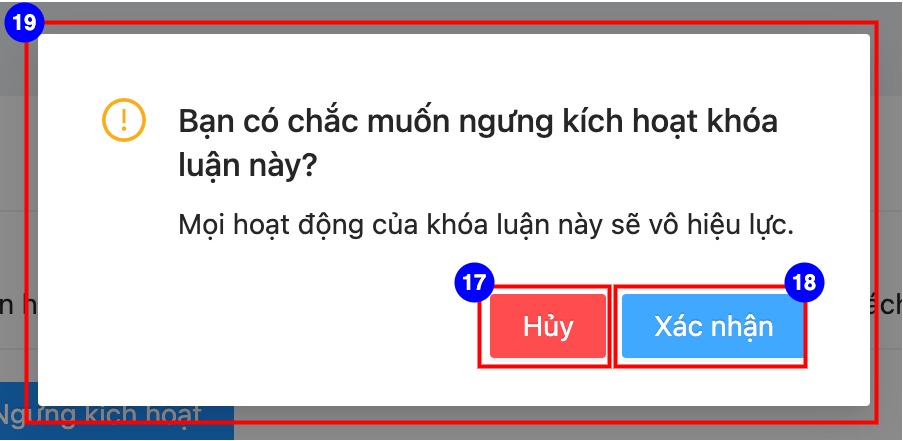


Figure 4.7. Inactive thesis confirm dialog

Table 4.20. Thesis list tabscreen specification

|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Thesis detail tab | When the user clicks on Item-1, the thesis detail tab will be displayed. |
| Item-2 | Instructor list tab | When the user clicks on Item-2, the instructor list tab will be displayed. |
| Item-3 | Student list tab | When the user clicks on Item-3, the student list tab will be displayed. |
| Item-4 | Topic list tab | When the user clicks on Item-4, the topic list tab will be displayed. |
| Item-5 | Council list tab | - When the user clicks on Item-5, the council list tab will be displayed.  - When current phase of target thesis isn’t defense, Item-5 not displayed. |
| Item-6 | Edit thesis button | - When the target thesis has status is “Đang hoạt động”, Item-6 will not display.  - When the user clicks on Item-6, the user will be redirected to edit thesis screen. |
| Item-7 | Delete thesis button | - When the target thesis has status is “Đang hoạt động”, Item-7 will not display.  - When the user clicks on Item-7, the Item-13 will display. |
| Item-8 | Inactive thesis button | When the user clicks on Item-7, the Item-19 will display. |
| Item-9 | Thesis detail information |  |
| Item-10 | Active thesis button | When the user clicks on Item-10, the Item-16 will display. |
| Item-11 | Cancel delete button | When the user clicks on Item-11, the Item-13 will close. |
| Item-12 | Submit delete button | When the user clicks on Item-12, the target thesis will be deleted and redirect the user to thesis list screen. |
| Item-13 | Delete thesis confirm dialog |  |
| Item-14 | Cancel active button | When the user clicks on Item-14, the Item-16 will close. |
| Item-15 | Submit active button | When the user clicks on Item-15, the target thesis is activated, and the Item-16 will close. |
| Item-16 | Active thesis confirm dialog |  |
| Item-17 | Cancel inactive button | When the user clicks on Item-17, the Item-19 will close. |
| Item-18 | Submit inactive button | When the user clicks on Item-18, the target thesis is inactivated, and the Item-19 will close. |
| Item-19 | Inactive thesis confirm dialog |  |

##### Instructor list tab

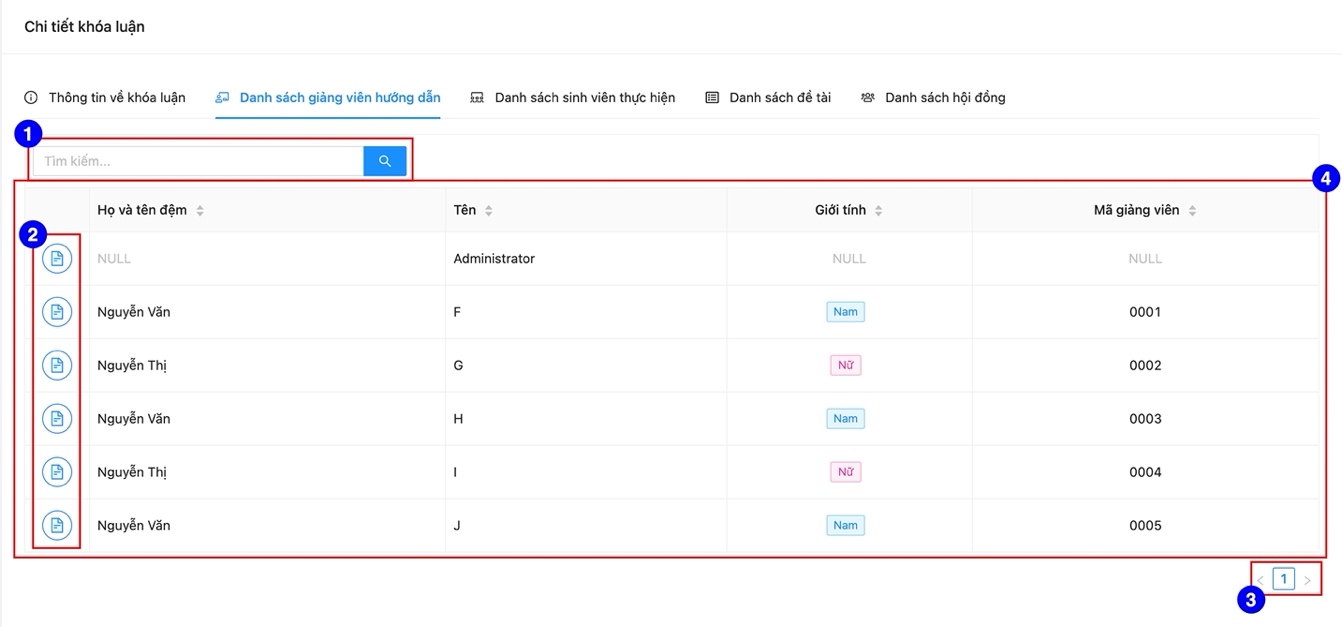


Figure 4.8. Instructor list tab

Table 4.21. Instructor list tabscreen specification

|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Search bar | When the user input keyword and clicks glass icon, search result will be displayed on Item-4. |
| Item-2 | Detail instructor button | When the user clicks on Item-2, the user will be redirected to corresponding detail instructor page. |
| Item-3 | Pagination button | When the user clicks on Item-3, the lecturer list will be displayed on Item-4 by page number. |
| Item-4 | Instructor list |  |

##### Student list tab

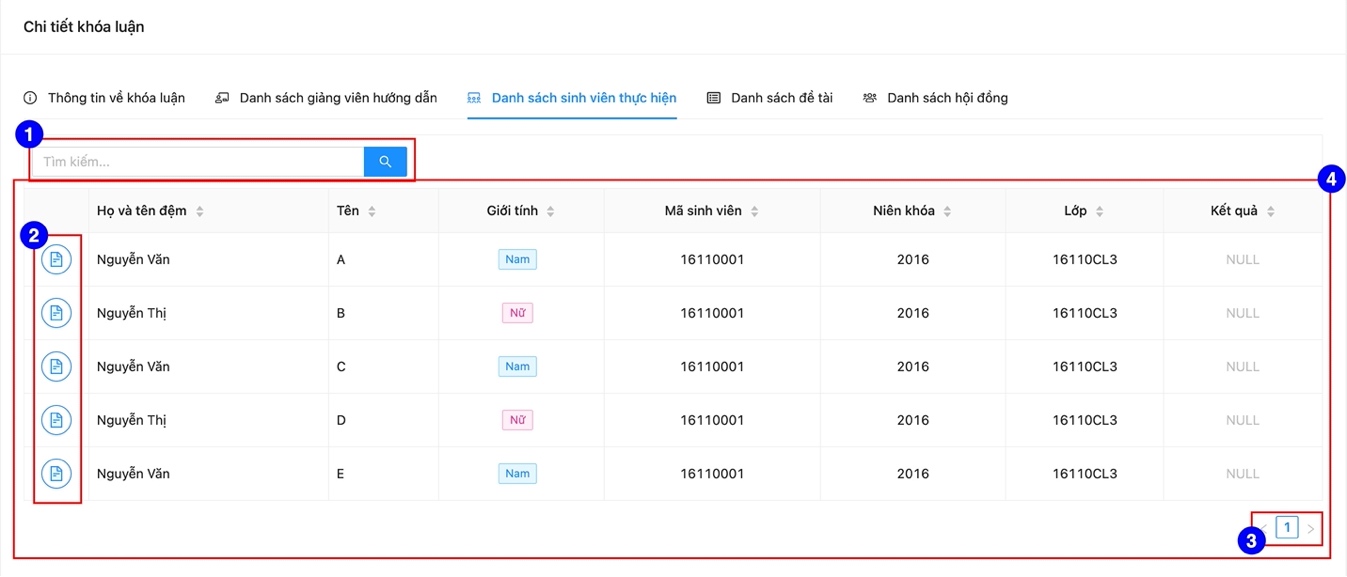


Figure 4.9. Student list tab

Table 4.22. Student list tabscreen specification

|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Search bar | When the user input keyword and clicks glass icon, search result will be displayed on Item-4. |
| Item-2 | Detail student button | When the user clicks on Item-2, the user will be redirected to corresponding detail student page. |
| Item-3 | Pagination button | When the user clicks on Item-3, the student list will be displayed on Item-4 by page number. |
| Item-4 | Instructor list |  |

##### Topic list tab

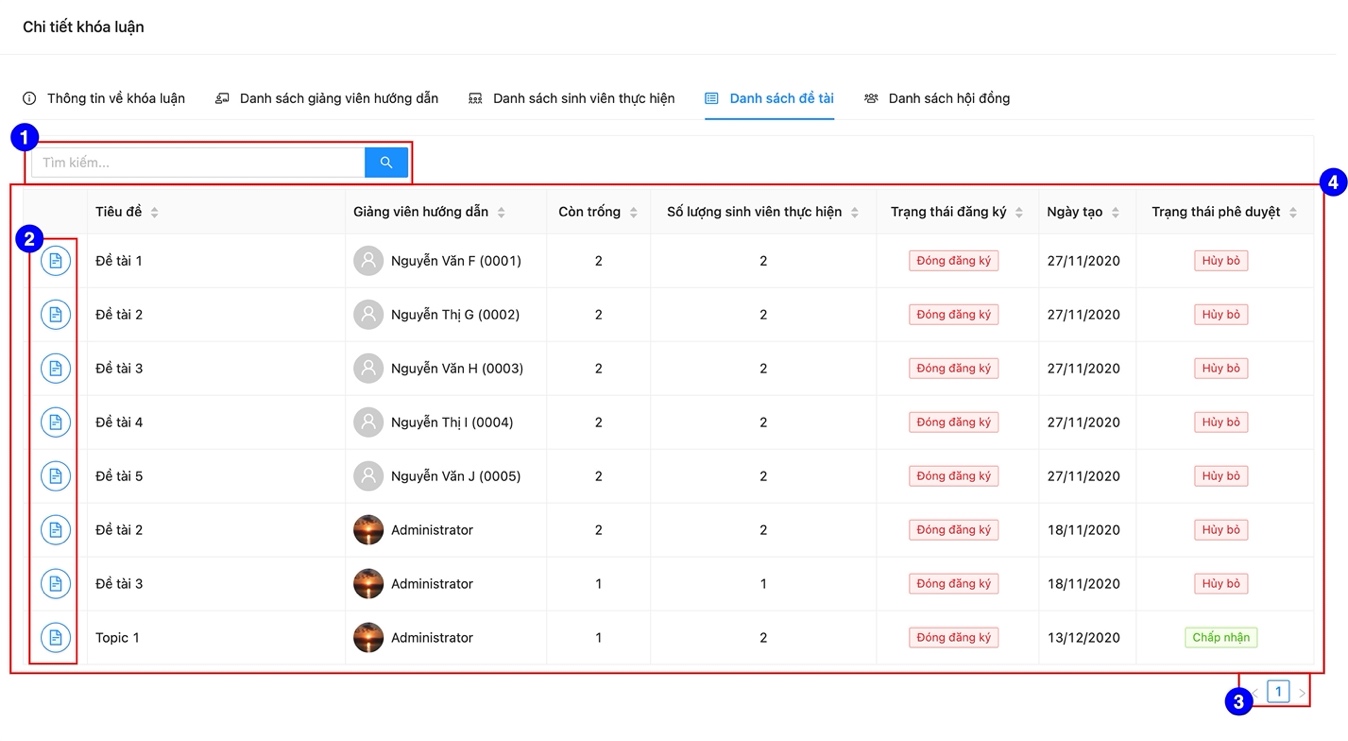


Figure 4.10. Topic list tab

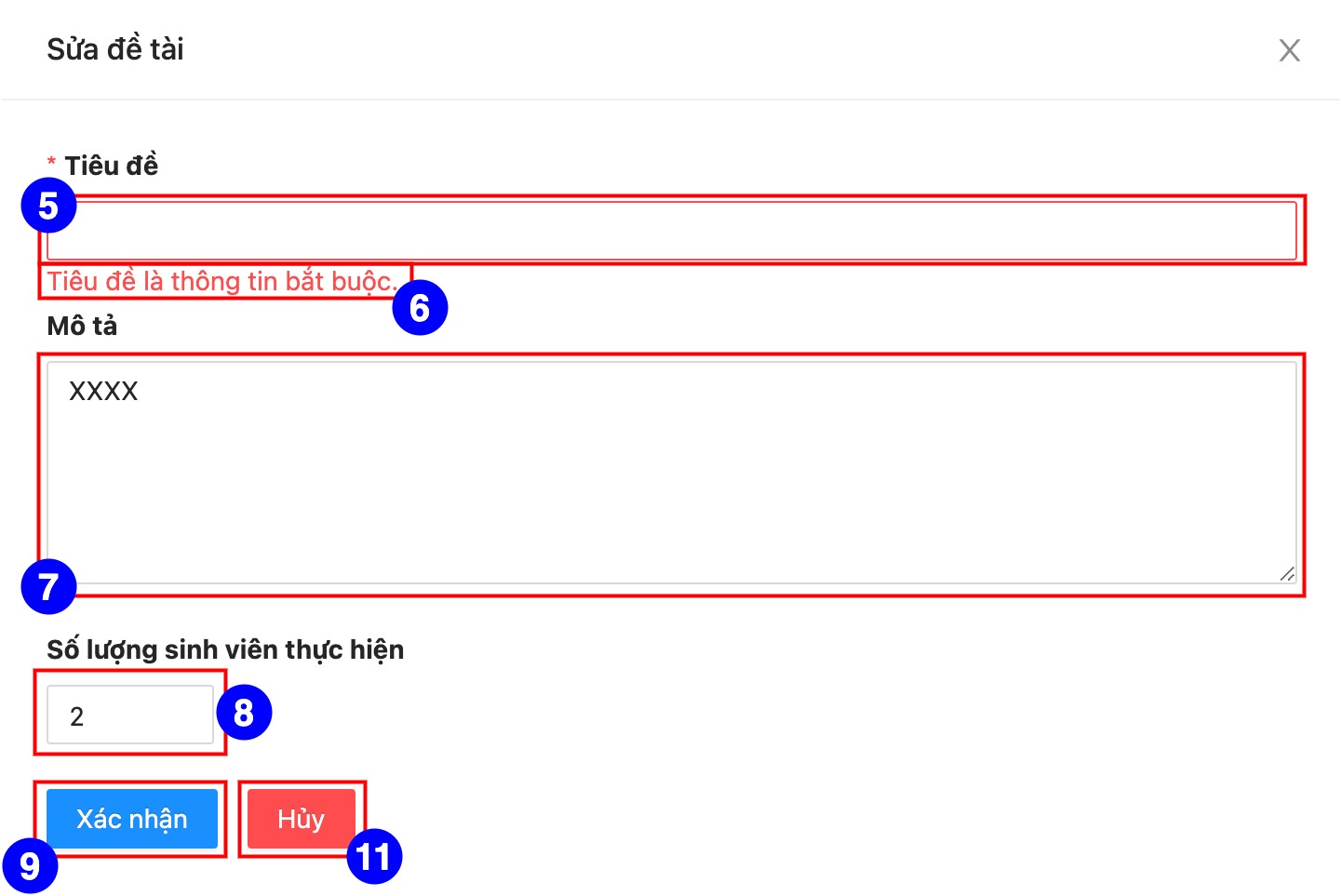


Figure 4.11. Create topic drawer

Table 4.23. Topic list tab specification

|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Search bar | When the user input keyword and clicks glass icon, search result will be displayed on Item-4. |
| Item-2 | Detail topic button | When the user clicks on Item-2, the user will be redirected to corresponding detail topic page. |
| Item-3 | Pagination button | When the user clicks on Item-3, the topic list will be displayed on Item-4 by page number. |
| Item-4 | Instructor list |  |
| Item-5 | Subject input field |  |
| Item-6 | Validation error message | When the user input invalid value or empty value on Item-5, Item-6 will display. |
| Item-7 | Description input field |  |
| Item-8 | Maximum student | - Minimum value is 1.  - Maximum value is 2 |
| Item-9 | Submit button | When the user clicks on Item-9, the create topic drawer will disappear. |
| Item-10 | Cancel button | When the user clicks on Item-10, the create topic drawer will disappear. |

##### Council list tab

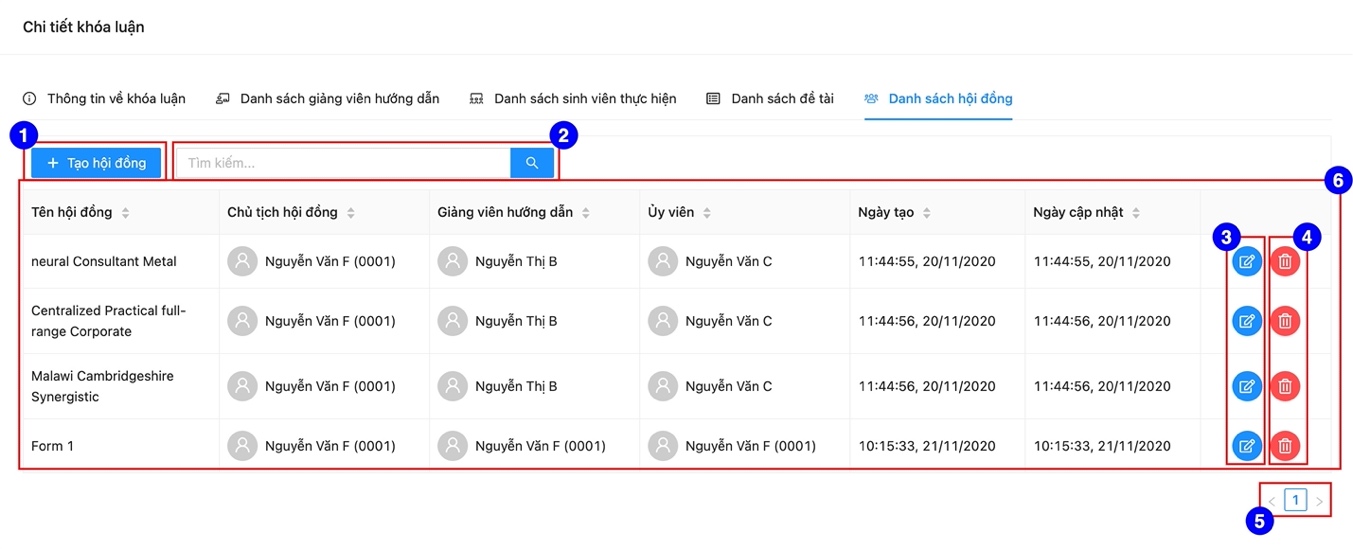
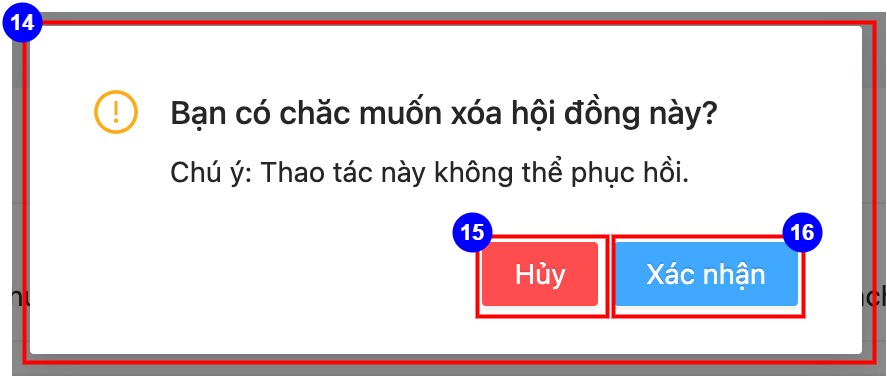


Figure 4.12. Council list tab

Graphical user interface, application

Description automatically generated

**Figure 4.13**. Create council



**Figure 4.14**. Delete council confirm dialog

Table 4.24. Council list tab specification

|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Create council button | When the user clicks on Item-1, Item-7 will display. |
| Item-2 | Search council bar | When the user input keyword and clicks glass icon, search result will be displayed on Item-6. |
| Item-3 | Edit council button | When the user clicks on Item-2, edit council drawer will display with UI similar Item-7. |
| Item-4 | Delete council button | When the user clicks on Item-4, Item-14 will display. |
| Item-5 | Pagination button | When the user clicks on Item-5, the thesis list will be displayed on Item-6 by page number. |
| Item-6 | Council list |  |
| Item-7 | Create council drawer |  |
| Item-8 | Council information input fields | When the user inputs invalid value or empty value, Item-9 will display |
| Item-9 | Validation error message |  |
| Item-10 | Submit button | When the user clicks on Item-10, Item-7 will disappear, and success notification will display. |
| Item-11 | Cancel button | When the user clicks on Item-10, Item-7 will disappear. |
| Item-12 | Chairman input field | When the user inputs value, Item-13 will display. |
| Item-13 | Suggestion list | When the user clicks on Item-13, target value will be displayed at Item-12 and Item-13 will disappear. |
| Item-14 | Delete council confirm dialog |  |
| Item-15 | Cancel delete confirm | When the user clicks on Item-15, Item-14 will disappear. |
| Item-16 | Submit delete confirm | When the user clicks on Item-16, Item-14 will disappear, and success notification will display. |

#### Topic detail screen

##### Topic detail tab

Graphical user interface, application

Description automatically generated

**Graphical user interface, application, Teams

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Figure 4.15**. Detail topic tab

Graphical user interface, application

Description automatically generated

**Figure 4.16**. Change resiter status of topic confirm dialog

Table 4.25. Topic detail tab specification

|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Edit topic button | When the user input keyword and clicks glass icon, search result will be displayed on Item-4. |
| Item-2 | Delete topic button | When the user input keyword and clicks glass icon, search result will be displayed on Item-4. |
| Item-3 | Detail topic information |  |
| Item-4 | Open register button | - When target topic has state is “Chấp nhận” and status is “Đóng đăng ký”.  - When the user clicks on Item-4, the Item-14 will display. |
| Item-5 | Close register button | - When target topic has state is “Chấp nhận” and status is “Mở đăng ký”.  - When the user clicks on Item-5, the council list will be displayed on Item-6 by page number. |
| Item-6 | Close register confirm dialog |  |
| Item-7 | Cancel close register button | When the user clicks on Item-7, Item-6 will disappear. |
| Item-8 | Submit close register button | When the user clicks on Item-8, Item-6 will disappear, and success notification will display. |
| Item-9 | Open register confirm dialog |  |
| Item-10 | Cancel open register button | When the user clicks on Item-10, Item-9 will disappear. |
| Item-11 | Submit open register button | When the user clicks on Item-11, Item-9 will disappear, and success notification will display. |

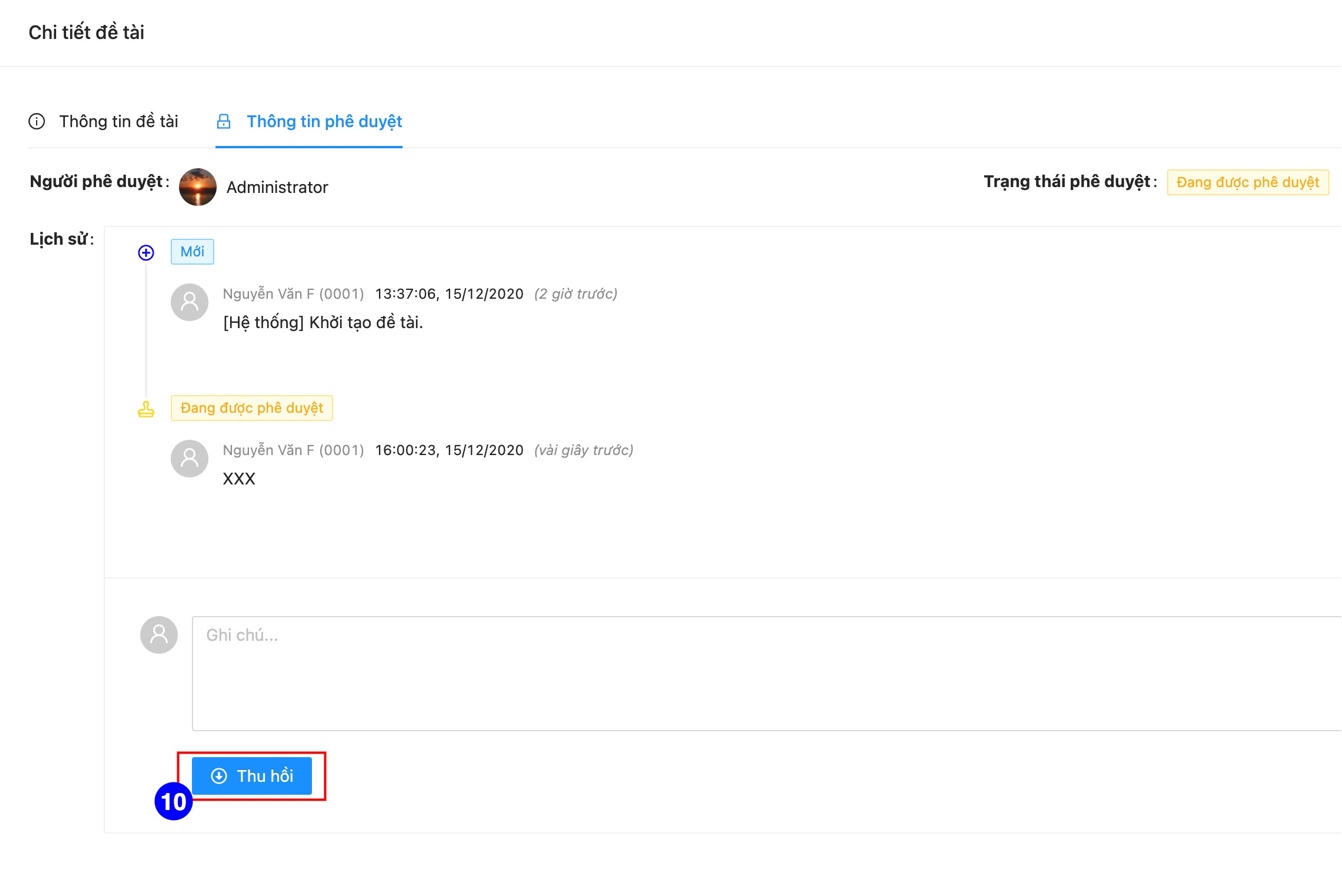
##### Approval information tab

**Graphical user interface, application, Teams

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

****

**Figure 4.17**. Approval information tab

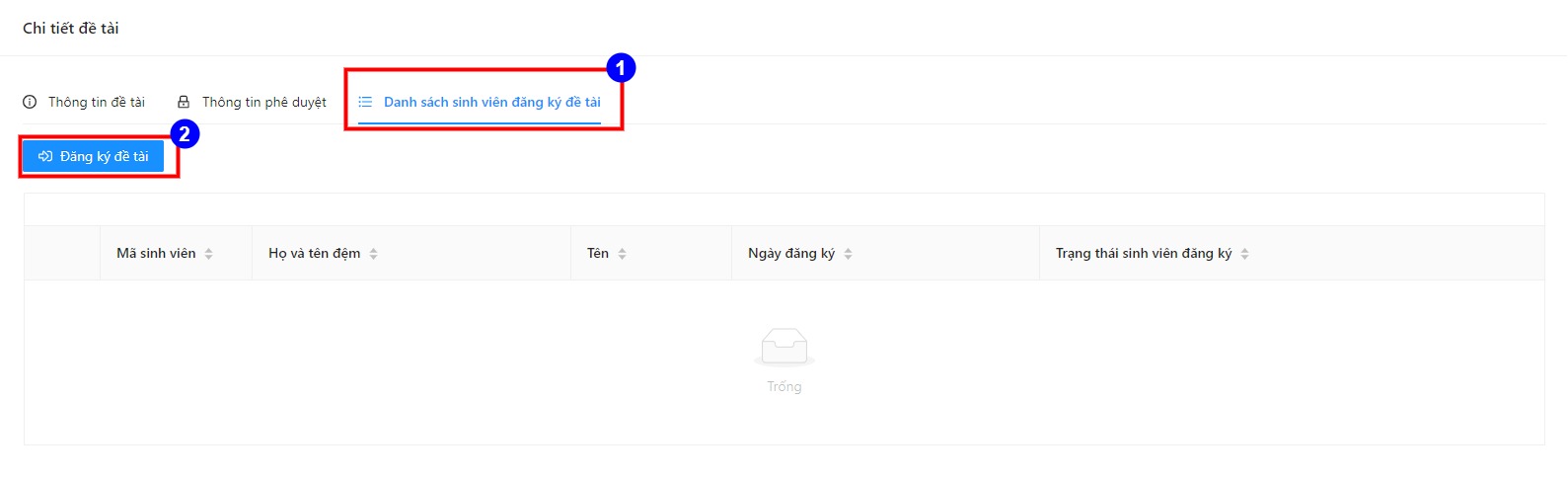


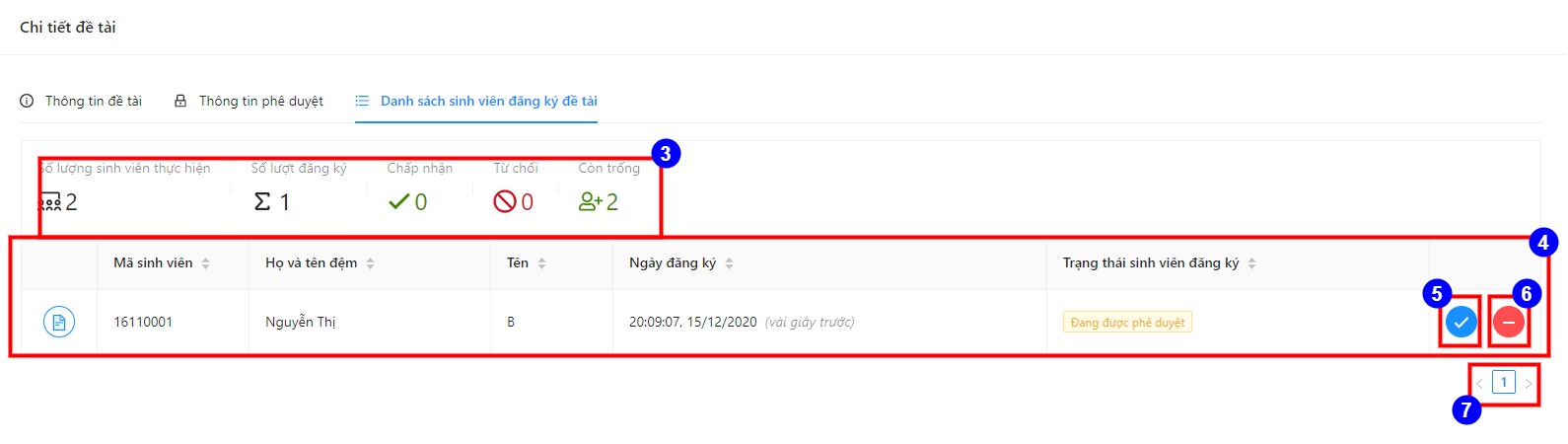
**Figure 4.18**. Confirm action dialog

Table 4.26. Approval information tab specification

|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Approver information |  |
| Item-2 | Approval status |  |
| Item-3 | Approval history |  |
| Item-4 | Action note input field |  |
| Item-5 | Send approval request button | - When the approval status of target topic is “Mới” hoặc “Trả lại” hoặc “Thu hồi”, Item-5 will display.  - When the user clicks on Item-5, Item-3 will be updated, and Item-2 will change to “Đang được phê duyệt”. |
| Item-6 | Cancel | - When the approval status of target topic is “Mới” hoặc “Trả lại” hoặc “Thu hồi”, Item-6 will display.  - When the user clicks on Item-6, Item-3 will be updated, and Item-2 will change to “Hủy bỏ”. |
| Item-7 | Accept button | - When the approval status of target topic is “Đang được phê duyệt”, Item-7 will display.  - When the user clicks on Item-7, Item-3 will be updated, and Item-2 will change to “Chấp nhận”. |
| Item-8 | Sendback button | - When the approval status of target topic is “Đang được phê duyệt”, Item-8 will display.  - When the user clicks on Item-8, Item-3 will be updated, and Item-2 will change to “Trả lại”. |
| Item-9 | Reject button | - When the approval status of target topic is “Đang được phê duyệt”, Item-9 will display.  - When the user clicks on Item-9, Item-3 will be updated, and Item-2 will change to “Từ chối”. |
| Item-10 | Withdraw button | - When the approval status of target topic is “Đang được phê duyệt”, Item-10 will display.  - When the user clicks on Item-10, Item-3 will be updated, and Item-2 will change to “Thu hồi”. |
| Item-11 | Confirm action dialog |  |
| Item-12 | Cancel confirm button | When the user clicks on Item-12, Item-11 will disappear. |
| Item-13 | Submit confirm button | When the user clicks on Item-13, Item-11 will disappear and success notificaion will display. |

##### Register list tab

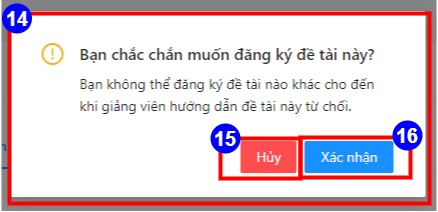
****

****

**Figure 4.19**. Register list tab



**Figure 4.20**. Register confirm dialog

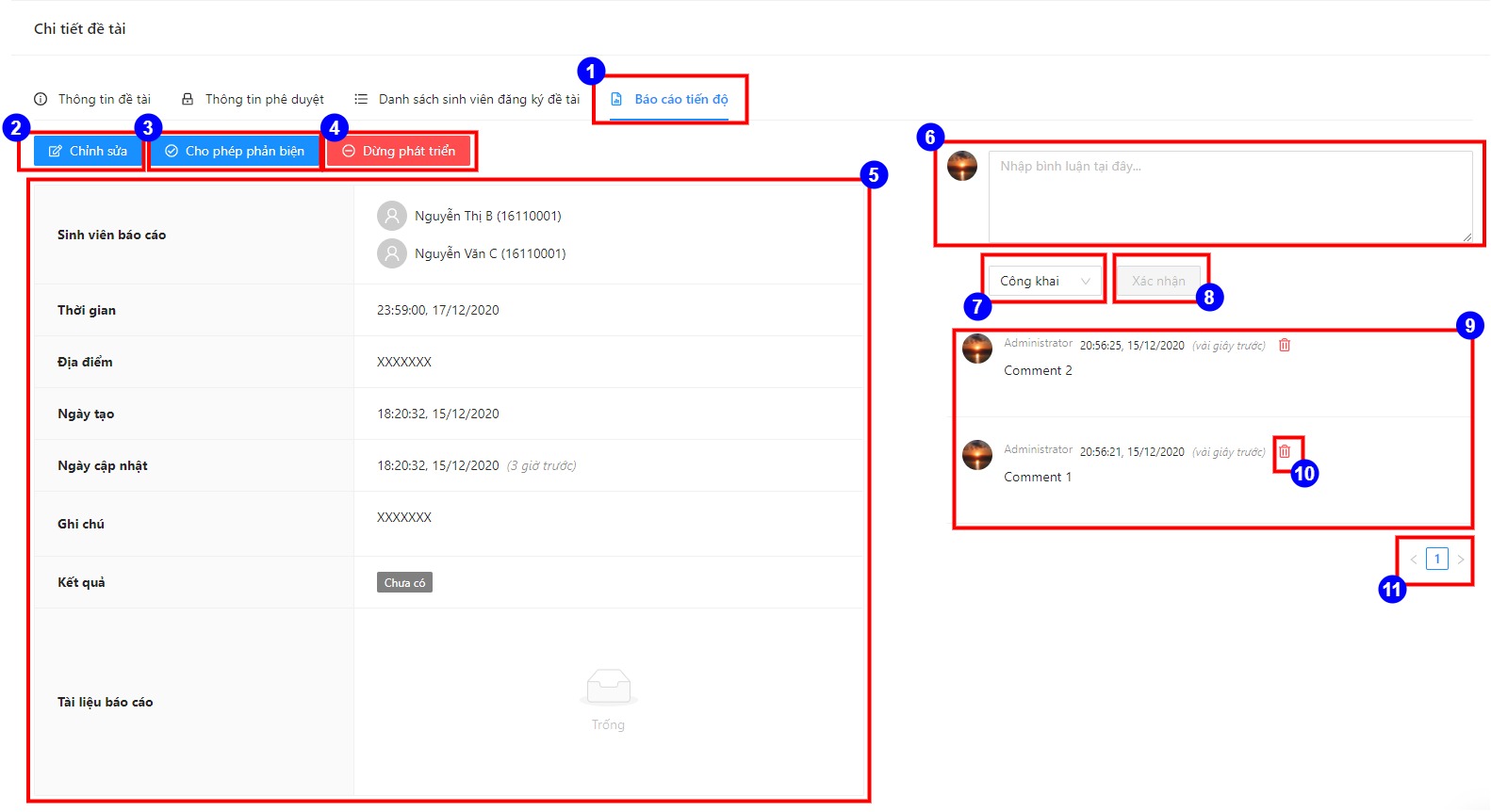


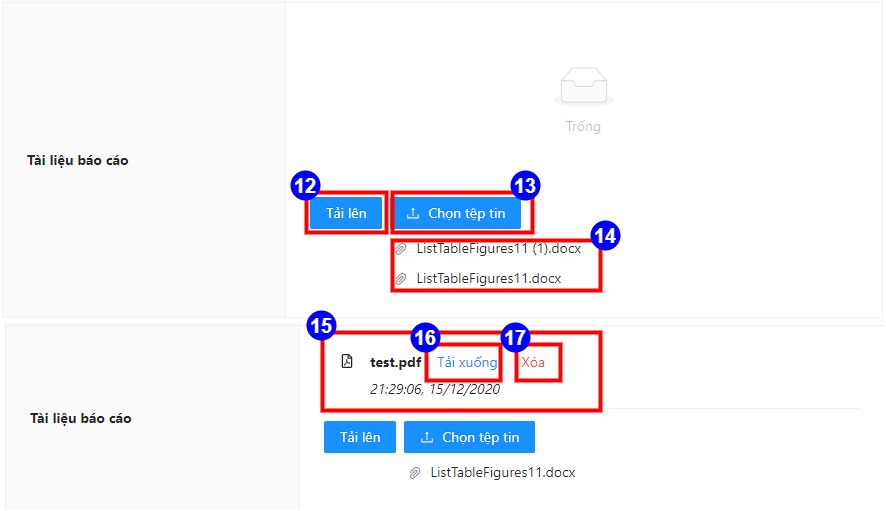
**Figure 4.21**. Register confirm dialog

Table 4.27. Register list tab specification

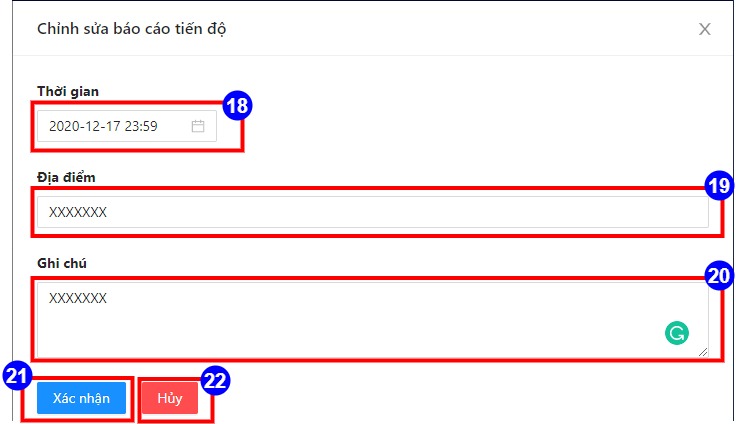
|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Register list tab button | - When the phase of target thesis is register for student and the status of target topic is “Chấp nhận”, Item-1 will display.  - When the user clicks on Item-1, register list tab will display. |
| Item-2 | Register button | - When login user is student and target topic has register status is open, Item-2 will display.  - When the user clicks on Item-2, Item-14 will display. |
| Item-3 | Topic register statistics |  |
| Item-4 | Register list |  |
| Item-5 | Accept register button | - When login user is owner of target topic, Item-5 will display.  - When the user clicks on Item-5, Item-8 will display. |
| Item-6 | Reject register button | - When login user is owner of target topic, Item-6 will display.  - When the user clicks on Item-6, Item-11 will display. |
| Item-7 | Pagination button | When the user clicks on Item-7, the register list will be displayed on Item-4 by page number. |
| Item-8 | Accept confirm dialog |  |
| Item-9 | Cancel accept confirm | When the user clicks on Item-9, Item-8 will disappear. |
| Item-10 | Submit accept confirm | When the user clicks on Item-10, Item-8 will disappear, and success notification will display. |
| Item-11 | Reject confirm dialog |  |
| Item-12 | Cancel reject confirm | When the user clicks on Item-12, Item-11 will disappear. |
| Item-13 | Submit reject confirm | When the user clicks on Item-13, Item-11 will disappear, and success notification will display. |
| Item-14 | Register confirm dialog |  |
| Item-15 | Cancel register confirm | When the user clicks on Item-15, Item-14 will disappear. |
| Item-16 | Submit register confirm | When the user clicks on Item-16, Item-14 will disappear, and success notification will display. |

##### Semi-progress report tab

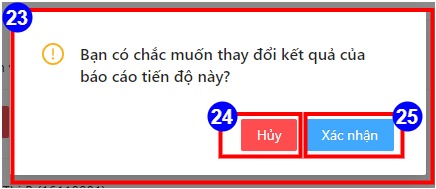
****

****

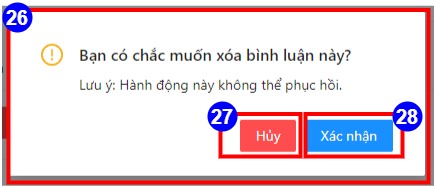
**Figure 4.22**. Semi-progress report tab



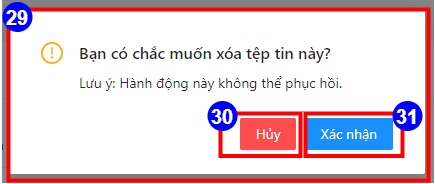
**Figure 4.23**. Edit semi-progress information drawer



**Figure 4.24**. Change result confirm dialog



**Figure 4.25**. Delete comment confirm dialog



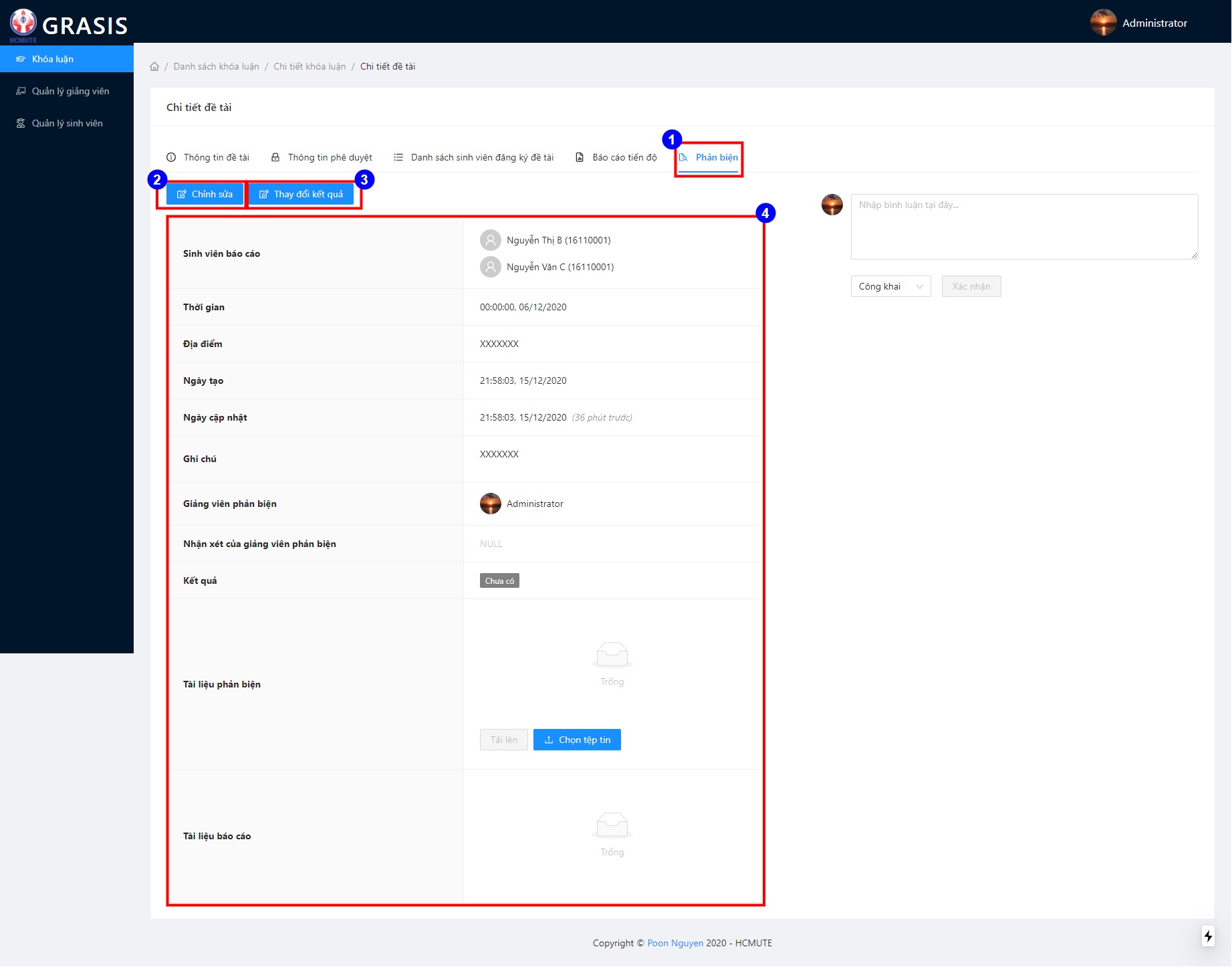
**Figure 4.26**. Delete document report confirm dialog

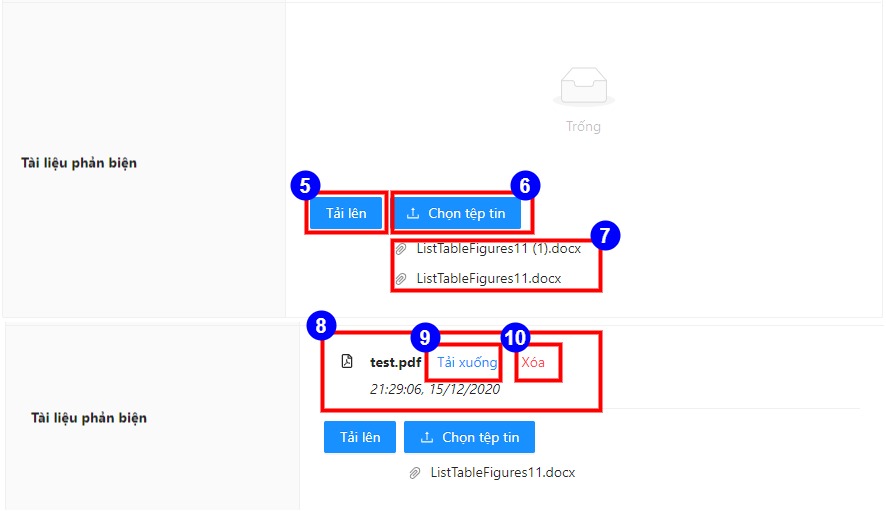
Table 4.28. Semi-progress report tab specification

|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Semi-progress report tab button | - When the phase of target thesis is semi-progress report and the status of target topic is “Chấp nhận”, Item-1 will display.  - When the user clicks on Item-1, semi-progress report tab will display. |
| Item-2 | Edit semi-progress report information button | - When login user is admin and the current phase of target topic is semi-progress report, Item-2 will display.  - When target semi-progress report has result, Item-2 will not display.  - When the user clicks on Item-2, the edit semi-progress information drawer will display. |
| Item-3 | Semi-progress detail information | When target semi-progress report has result, Item-3 will not display. |
| Item-4 | Accept review button | - When target semi-progress report has result, Item-4 will not display.  - When the user clicks on Item-4, Item-20 will display |
| Item-5 | Reject review button | When the user clicks on Item-5, Item-20 will display. |
| Item-6 | Comment input field |  |
| Item-7 | Comment types selector | - When loggin user is student, Item-7 not display.  - When the user clicks on Item-7, a comment types will display, including:   * Private type * Public type |
| Item-8 | Post comment button | When the user clicks on Item-8, Item-6 will empty, and Item-9 will be updated. |
| Item-9 | Comment list |  |
| Item-10 | Delete comment button | When the user clicks on Item-10, Item-26 will display. |
| Item-11 | Comment pagination button | When the user clicks on Item-11, the comment list will be displayed on Item-9 by page number. |
| Item-12 | Submit upload | When the user clicks on Item-12, Item-14 disappear. |
| Item-13 | Select upload file button | When the user clicks on Item-13, file upload dialog will display. |
| Item-14 | Upload file list |  |
| Item-15 | Uploaded file list |  |
| Item-16 | Download uploaded file button | When the user clicks on Item-16, file will be downloaded. |
| Item-17 | Delete uploaded file button | - When loggin user is student, Item-17 will display.  - When the user clicks on Item-17, Item-29 will display. |
| Item-18 | Datetime input field | When the user clicks on Item-18, datetime dialog will display. |
| Item-19 | Place input field |  |
| Item-20 | Note input field |  |
| Item-21 | Submit edit button | When the user clicks on Item-21, drawer will close, and success notification will display. |
| Item-22 | Cancel edit button | When the user clicks on Item-22, drawer will close. |
| Item-23 | Change result confirm dialog |  |
| Item-24 | Cancel change result confirm dialog | When the user clicks on Item-24, Item-23 will disappear. |
| Item-25 | Submit change result confirm dialog | When the user clicks on Item-25, Item-23 will disappear, and success notification will display. |
| Item-26 | Delete comment confirm dialog |  |
| Item-27 | Delete comment confirm dialog | When the user clicks on Item-27, Item-26 will disappear. |
| Item-28 | Submit delete comment confirm dialog | When the user clicks on Item-28, Item-26 will disappear, and success notification will display. |
| Item-29 | Delete uploaded file confirm dialog |  |
| Item-30 | Delete uploaded file confirm dialog | When the user clicks on Item-30, Item-27 will disappear. |
| Item-31 | Submit delete uploaded file confirm dialog | When the user clicks on Item-31, Item-27 will disappear, and success notification will display. |

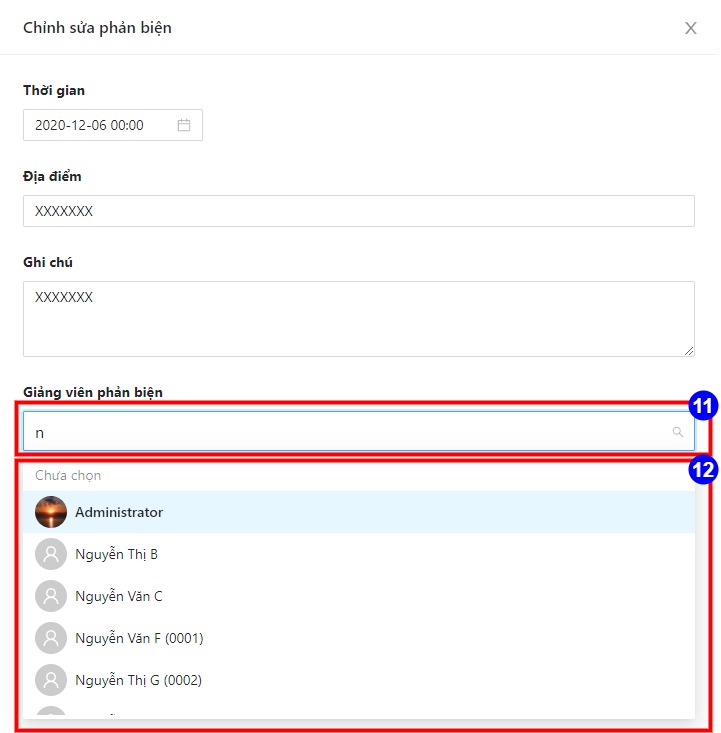
##### Review tab

In this section, items which similar items on above sections will not have specification.

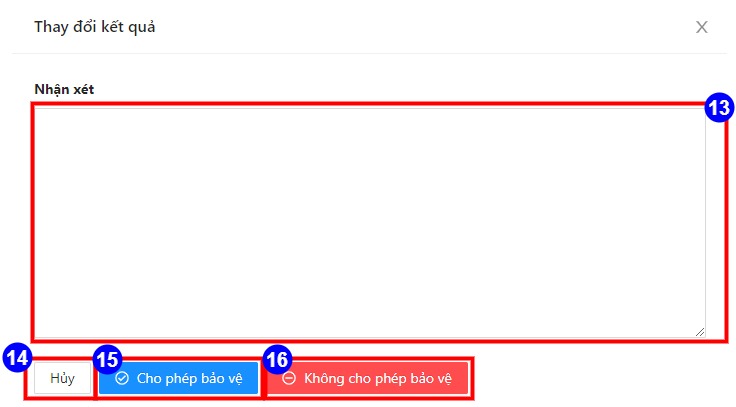
****

****

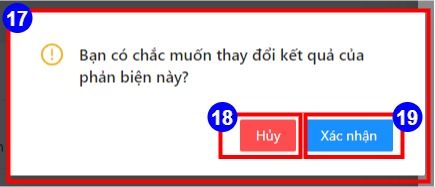
**Figure 4.27**. Review tab



**Figure 4.28**. Edit review drawer



**Figure 4.29**. Change result confirm drawer



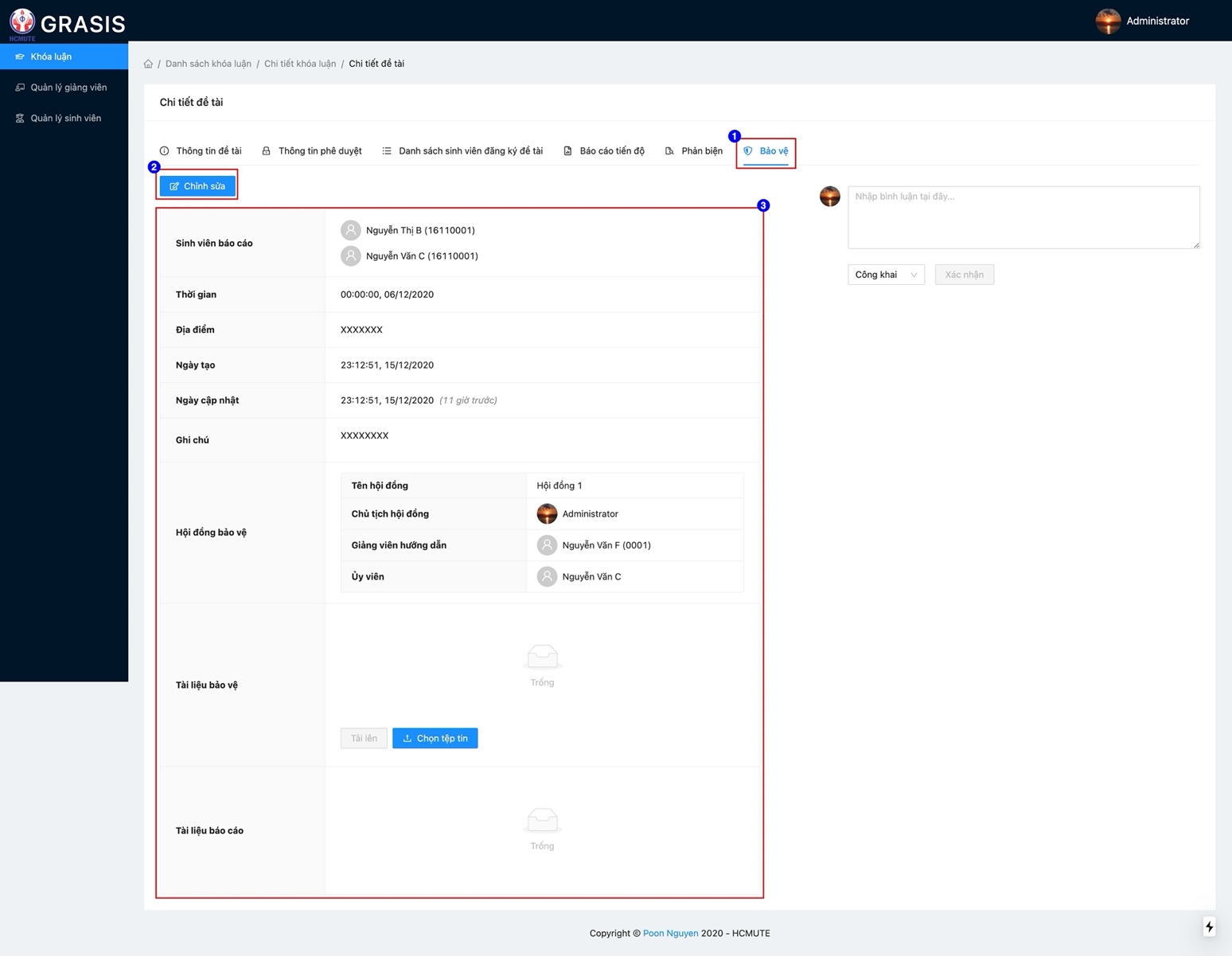
**Figure 4.30**. Change result confirm dialog

Table 4.29. Review tab specification

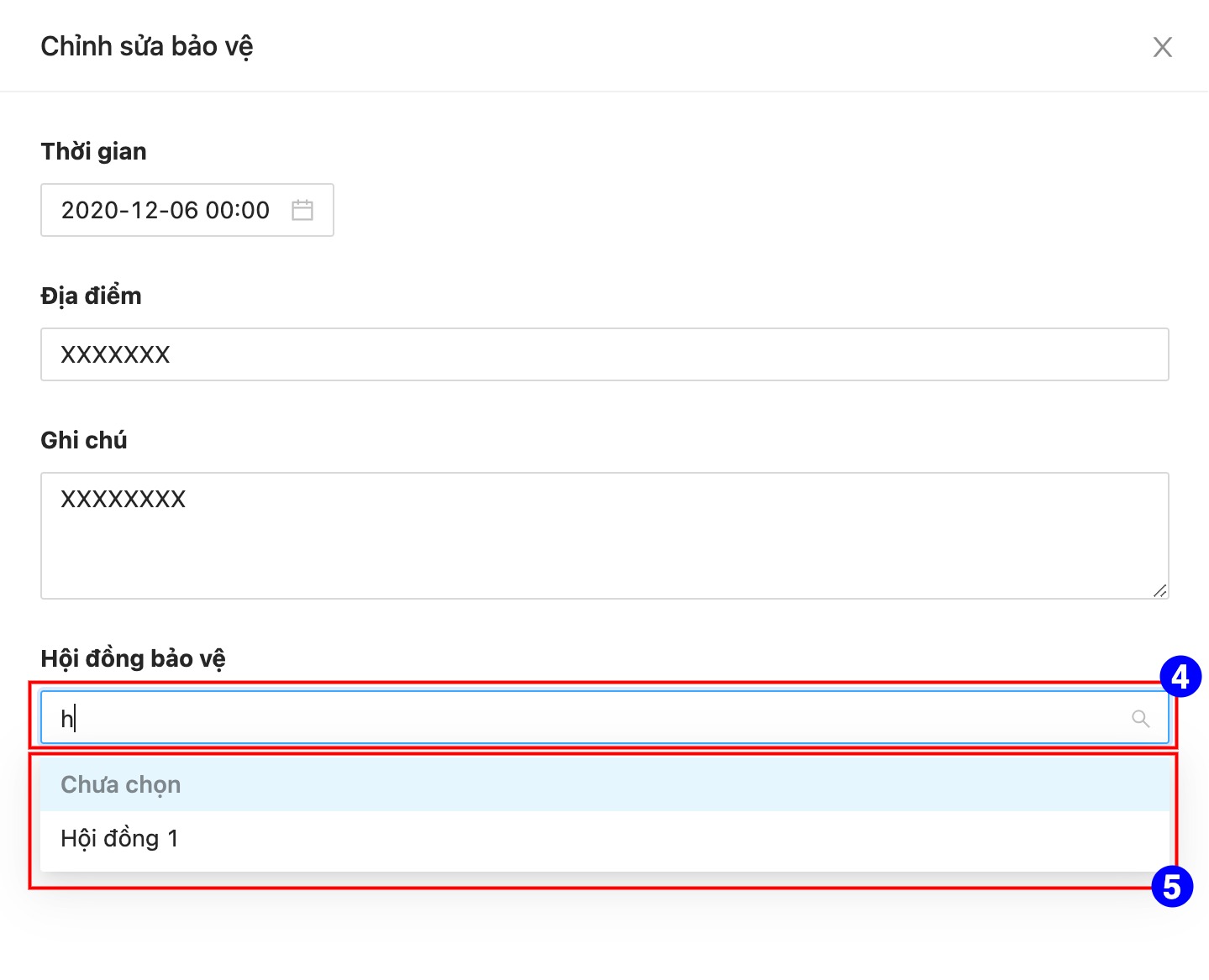
|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Review tab button | - When the phase of target thesis is review and the status of target topic is “Chấp nhận”, Item-1 will display.  - When the user clicks on Item-1, review tab will display. |
| Item-2 | Edit review information button | - When login user is admin and the current phase of target topic is review, Item-2 will display.  - When target review has result, Item-2 will not display.  - When the user clicks on Item-2, the edit review information drawer will display. |
| Item-3 | Change result button | When the user clicks on Item-3, the change result drawer will display. |
| Item-4 | Review detail information |  |
| Item-5 | Submit upload | - When loggin user is reviewer, Item-5 will display.  - When the user clicks on Item-12, Item-14 disappear. |
| Item-6 | Select upload file button | - When loggin user is reviewer, Item-5 will display.  - When the user clicks on Item-13, file upload dialog will display. |
| Item-7 | Upload file list |  |
| Item-8 | Uploaded file list |  |
| Item-9 | Download uploaded file button | When the user clicks on Item-16, file will be downloaded. |
| Item-10 | Delete comment button | - When loggin user is reviewer, Item-5 will display.  - When the user clicks on Item-10, the delete confirm dialog will display. |
| Item-11 | Reviewer input field | When the user inputs value, Item-12 will display. |
| Item-12 | Suggest reviewers | When the user clicks on Item-12, Item-12 disappear, and value display on Item-11. |
| Item-13 | Comment of reviewer input field |  |
| Item-14 | Cancel change result | When the user clicks on Item-14, the drawer will disappear. |
| Item-15 | Allow defense button | When the user clicks on Item-15, Item-17 will display |
| Item-16 | Disallow defense button | When the user clicks on Item-15, Item-18 will display. |
| Item-17 | Change result confirm dialog |  |
| Item-18 | Cancel change result confirm dialog | When the user clicks on Item-18, Item-17 will disappear. |
| Item-19 | Submit change result confirm dialog | When the user clicks on Item-19, Item-17 will disappear, and success notification will display. |

##### Defense tab

In this section, items which similar items on above sections will not have specification.

****

**Figure 4.31**. Defense tab

****

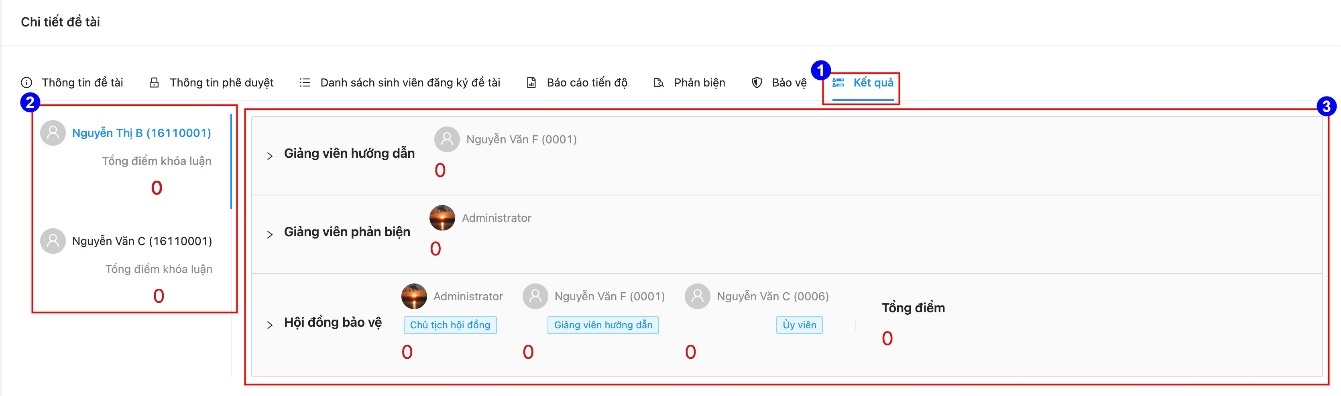
**Figure 4.32**. Edit defense drawer

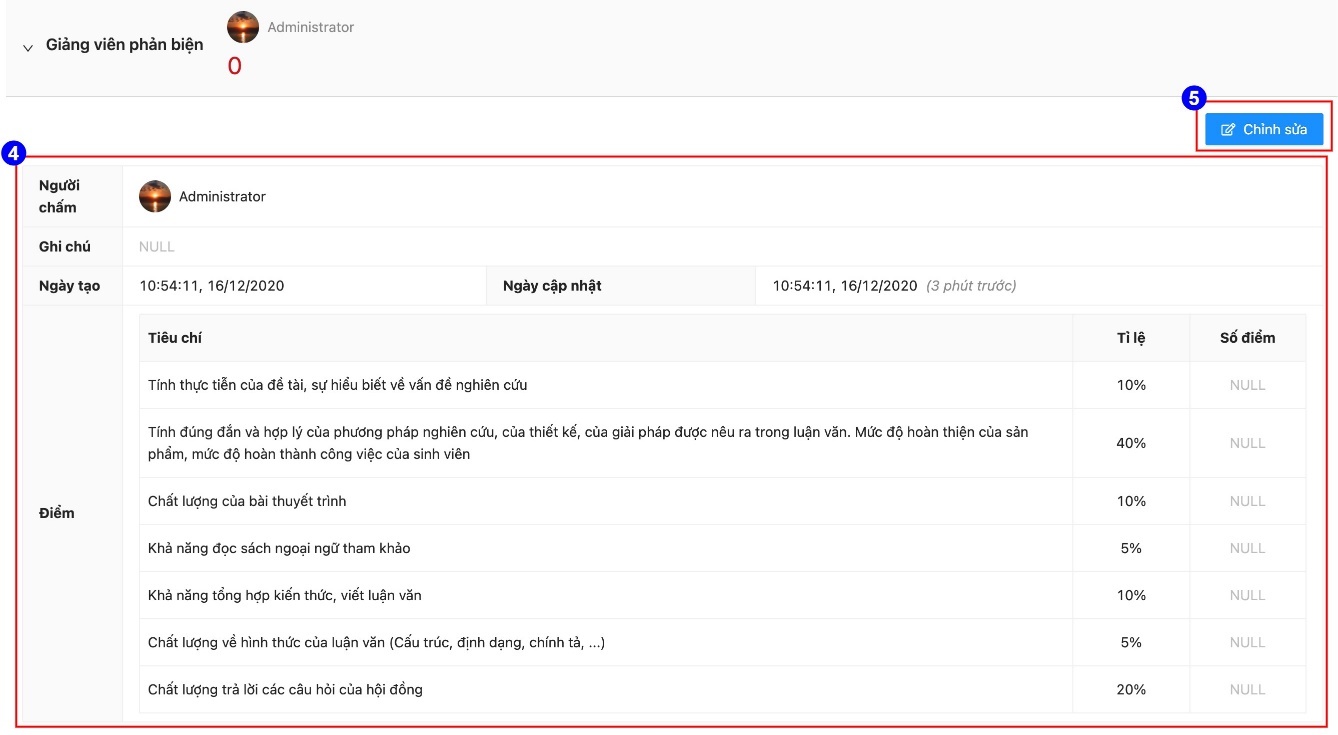
Table 4.30. Defense tab specification

|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Defense tab button | - When the phase of target thesis is defense and the status of target topic is “Chấp nhận”, Item-1 will display.  - When the user clicks on Item-1, defense tab will display. |
| Item-2 | Edit defense information button | - When login user is admin and the current phase of target topic is defense, Item-2 will display.  - When the user clicks on Item-2, the edit defense information drawer will display. |
| Item-3 | Review detail information |  |
| Item-4 | Defense council input field | When the user input value on Item-4, Item-5 will display. |
| Item-5 | Suggest defense councils | When the user clicks on Item-5, Item-5 disappear, and value display on Item-4. |

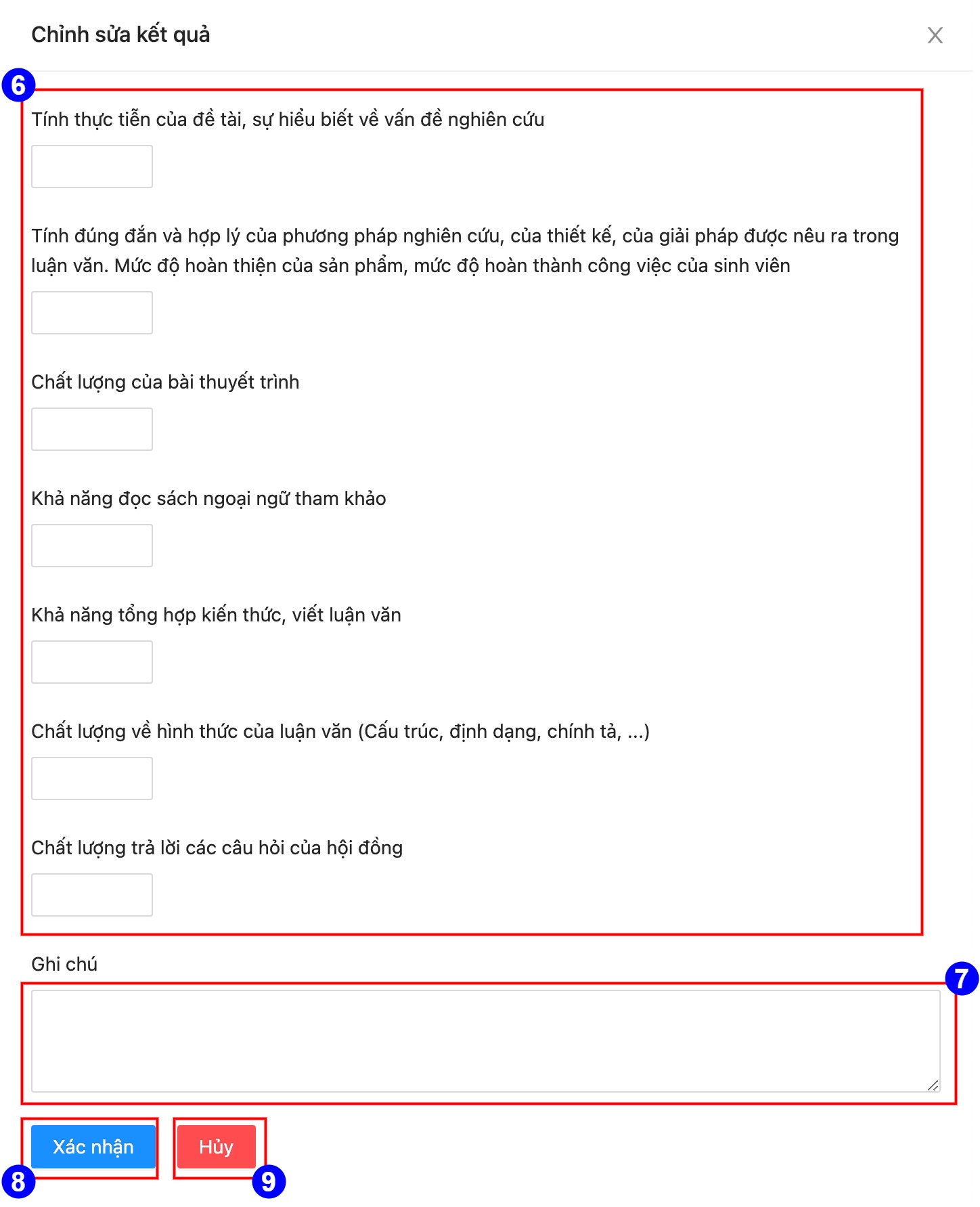
##### Result tab

In this section, items which similar items on above sections will not have specification.

****

****

**Figure 4.33**. Result tab

****

**Figure 4.34**. Edit result drawer

Table 4.31. Result tab specification

|  |  |  |
| --- | --- | --- |
| Item | Descriptions | Operations |
| Item-1 | Result tab button | - When the phase of target thesis is publish result and the status of target topic is “Chấp nhận”, Item-1 will display.  - When the user clicks on Item-1, result tab will display. |
| Item-2 | Student list | When the user clicks on Item-2, Item-3 will update by target student. |
| Item-3 | Result list | When the user clicks on Item-3, Item-4 will display by target result. |
| Item-4 | Result detail information | When loggin user is not owner of result, specify result will have “NULL” value (except summary result). |
| Item-5 | Edit result button | - When loggin user is not owner of result, Item-5 will not display.  - When the user clicks on Item-5, the edit result drawer will display. |
| Item-6 | The field enters the results according to each criterion |  |
| Item-7 | Note input field |  |
| Item-8 | Submit edit result button | When the user clicks on Item-8, drawer will disappear, and success notification will display. |
| Item-9 | Cancel edit result button | When the user clicks on Item-9, drawer will disappear. |

# **CHAPTER 5.** **INSTALLATION AND TESTING**

## **Installation**

Download source code from: https://github.com/doonpy/grasis

### Production environment

#### Libraries and software need

Table 5.1. Libraries and software need to be installed in production

|  |  |  |
| --- | --- | --- |
| No | Library/Software | Download Url |
| 1 | NodeJS 14.x or later (include Yarn and NPM) | https://nodejs.org |
| 2 | MySQL Community Server 5.7 | https://downloads.mysql.com/archives/community/ |
| 3 | Redis 6.0.9 or later | https://redis.io |

#### Step by step to deploy

* Step 1: Crete database with name “grasis” in MySQL Community Server.
* Step 2:Open file **api/config/production.env** and config following params:
  + JAWSDB\_URL: Database connection string.
  + REDISCLOUD\_URL: Redis connection string.
  + AWS\_REGION: AWS region (default us-east-1).
  + AWS\_ACCESS\_KEY\_ID: AWS access key for AWS S3 service.
  + AWS\_SECRET\_ACCESS\_KEY: AWS secret access key for AWS S3 service.
  + AWS\_BUCKET\_NAME: Bucket name of AWS S3 service.
* Step 3:Open file **web/.env.production** and config following params:
  + NEXT\_PUBLIC\_API\_SERVER: API sever address.
* Step 4: Run shell script file **deploy/production.sh** to deploy.

### Development environment

#### Libraries and software need

Table 5.2. Libraries and software need to be installed in development

|  |  |  |
| --- | --- | --- |
| No | Library/Software | Download Url |
| 1 | NodeJS 14.x or later (include Yarn and NPM) | https://nodejs.org |
| 2 | MySQL Community Server 5.7 | https://downloads.mysql.com/archives/community/ |
| 3 | Redis 6.0.9 or later | https://redis.io |
| 4 | NestJS CLI package | https://docs.nestjs.com/cli/overview |
| 5 | NextJS CLI package | https://nextjs.org/docs/api-reference/cli |

#### Step by step to configuration

* Step 1: Crete database with name “grasis” in MySQL Community Server.
* Step 2:Open file **api/config/local.env** and config following params:
  + JAWSDB\_URL: Database connection string.
  + REDISCLOUD\_URL: Redis connection string.
* Step 3:Open file **web/.env.production** and config following params:
  + NEXT\_PUBLIC\_API\_SERVER: API sever address.

#### Command cheatsheet

Table 5.3. List of command for install and run projects

|  |  |  |
| --- | --- | --- |
| No | Command | Description |
| 1 | yarn run install | Install dependencies. |
| 2 | yarn run build | Build source code to production code. |
| 3 | yarn --cwd api start:dev | Start API sever for development |
| 4 | yarn --cwd api start:prod | Start API server for production (require build command before). |
| 5 | yarn --cwd web start:dev | Start web application for development |
| 6 | yarn --cwd web start:prod | Start web application for production (require build command before). |

## 

## Testing

### Thesis

#### Function create thesis

Table 5.4. Test case function create thesis

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case Description | | Create thesis – Positive test case | | | Test Priority | | High |
| Pre-Requisite | | - Valid thesis information.  - Users has lecturer type.  - Users has student type. | | | **Post-Requisite** | | NA |
| Test Execution Steps: | | | | | | | |
| No | **Action** | | **Inputs** | **Expected Output** | | **Test Result** | **Test Comments** |
| 1 | Login to system | | The correct username and passsword | Return to the account screen | | Passed |  |
| 2 | Click add thesis button | |  | Redirect to the create thesis screen | | Passed |  |
| 3 | Input thesis information | | The correct thesis information | Not display validation error messages | | Passed |  |
| 4 | Click “Xác nhận” button | |  | Thesis was created and redirect to thesis detail screen. | | Passed |  |

#### Function active thesis

Table 5.5. Test case function active thesis

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case Description | | Active thesis – Positive test case | | | Test Priority | | High |
| Pre-Requisite | | Thesis has status is inactive | | | **Post-Requisite** | | NA |
| Test Execution Steps: | | | | | | | |
| No | **Action** | | **Inputs** | **Expected Output** | | **Test Result** | **Test Comments** |
| 1 | Login to system | | The correct username and passsword | Return to the account screen | | Passed |  |
| 2 | Click on thesis detail button | |  | Redirect to the thesis detail screen | | Passed |  |
| 3 | Click on “Kích hoạt” button | |  | Confirm dialog display | | Passed |  |
| 4 | Click “Xác nhận” button | |  | Thesis was activated and confirm dialog disappear. | | Passed |  |

### Topic

#### Function create topic

Table 5.6. Test case function create topic

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case Description | | Create topic – Positive test case | | | Test Priority | | High |
| Pre-Requisite | | Valid topic information | | | **Post-Requisite** | | NA |
| Test Execution Steps: | | | | | | | |
| No | **Action** | | **Inputs** | **Expected Output** | | **Test Result** | **Test Comments** |
| 1 | Login to system | | The correct username and passsword | Return to the account screen | | Passed |  |
| 2 | Click on thesis detail button | |  | Redirect to the create thesis screen | | Passed |  |
| 3 | Click on topic list tab | |  | Topic list tab display | | Passed |  |
| 4 | Click “Tạo đề tài” button | |  | Create topic drawer display | | Passed |  |
| 5 | Input topic information | | The correct topic information | Not display validation error messages | | Passed |  |
| 6 | Click “Xác nhận” button | |  | Topic was created and redirect to topic detail screen | | Passed |  |

#### Function approval topic

Table 5.7. Test case function approval topic

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case Description | | Approval topic – Positive test case | | | Test Priority | | High |
| Pre-Requisite | | Topic has status is “Mới” | | | **Post-Requisite** | | NA |
| Test Execution Steps: | | | | | | | |
| No | **Action** | | **Inputs** | **Expected Output** | | **Test Result** | **Test Comments** |
| 1 | Login to system | | The correct username and passsword | Return to the account screen | | Passed |  |
| 2 | Click on thesis detail button | |  | Redirect to the thesis detail screen | | Passed |  |
| 3 | Click on topic list tab | |  | Topic list tab display | | Passed |  |
| 4 | Click on topic detail button | |  | Redirect to the topic detail screen | | Passed |  |
| 5 | Click on approval tab | |  | Approval tab display | | Passed |  |
| 6 | Click “Yêu cầu phê duyệt” button | |  | Confirm dialog display | | Passed |  |
| 7 | Click “Xác nhận” button | |  | Confirm dialog disappear and status of target topic change to “Đang được phê duyệt” | |  |  |

#### Function upload report document

Table 5.8. Test case function upload report document

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case Description | | Upload report document – Positive test case | | | Test Priority | | High |
| Pre-Requisite | | - Thesis has current phase is semi-progress report.  - Topic has status is “Chấp nhận” and has at least one student doing it.  - Semi-progress report of target topic has result is “Chưa có”.  - A file has extension is .doc or .pdf | | | **Post-Requisite** | | NA |
| Test Execution Steps: | | | | | | | |
| No | **Action** | | **Inputs** | **Expected Output** | | **Test Result** | **Test Comments** |
| 1 | Login to system | | The correct username and passsword | Return to the account screen | | Passed |  |
| 2 | Click on thesis detail button | |  | Redirect to the thesis detail screen | | Passed |  |
| 3 | Click on topic list tab | |  | Topic list tab display | | Passed |  |
| 4 | Click on topic detail button | |  | Redirect to the topic detail screen | | Passed |  |
| 5 | Click on semi-progress report tab | |  | Semi-progress report tab display | | Passed |  |
| 6 | Click “Chọn tệp tin” button | |  | File selector display | | Passed |  |
| 7 | Select file which is prepared | |  | Selected file display in selected file list | | Passed |  |
| 8 | Click “Tải lên” button | |  | Selected file was uploaded and success notification display | | Passed |  |

# CHAPTER 6. CONCLUSIONS AND DEVELOPMENT STRATEGY

## Results

The author researched and concluded the current graduate thesis process of the Faculty of High Quality Training. From that, understood and stated current inadequacies and building an information technology system to solve these.

Beside that, the author gets knowledge about the communication, authentication, and authorization process between API server and web application through REST APIs and JWT token.

Through building the server API, the author gains knowledge of database design, REST APIs, authentication, and authorization mechanisms. In addition, the author also understands the NestJS knowledge to build a stable API server with high scalability and good load balancing. In addition, for the server API to have high response speed, understanding and using Redis as a database caching has contributed significantly.

Moreover, the author also understands the knowledge of NextJS and applies it to build graduation thesis organization application on web platform. Application is based on the concept of SPA - Single Page Application makes the user experience better because it has limited the need to reload the page. With the support of the Static Generation Rendering (SGR) rendering engine, the responsiveness and performance of the application are always of good quality in any infrastructure condition. In addition, the application uses the Ant Design design library, a library commonly used in real-world websites. Since then, the interface of the application is friendly, easy to use and more eye-catching, but it takes less time to design and can be reused many times.

Finally, this project also gives me the knowledge of continuous integration (CI) and its use in project development. CI makes the development process faster and more stable. The change of source will affect the currently operating product, so applying CI in the development process helps early and timely detection of risks and errors that may occur when the product is put into operation. From there, it helps to save costs and manpower in software development.

## Restrictions

* Web application UX is still limited, making it inconvenient for users.
* Not yet supported on mobile devices.
* The source code is not optimized, and the processing logic is complicated.

## Development strategy

* Support on mobile devices.
* Improve UI/UX of web application.
* Implement new features such as notification, progress tracking between instructor and student, system settings, personal settings, import and export data, …
* Link to university database for easy management and synchronization.

# REFERENCES

[1] MySQL. (2020). Overview of the MySQL Database Management System. Retrieved from: https://dev.mysql.com/doc/refman/8.0/en/what-is.html [Accessed 07 Dec. 2020].

[2] Tutorialspoint. (2020). MySQL – Introduction. Retrieved from: https://www.tutorialspoint.com/mysql/mysql-introduction.htm [Accessed 07 Dec. 2020].

[3] Redis. (2020). Introduction to Redis. Retrieved from: https://redis.io/topics/introduction [Accessed 07 Dec. 2020].

[4] Wikipedia. (2020). Redis – History section. Retrieved from: https://en.wikipedia.org/wiki/Redis#History [Accessed 07 Dec. 2020].

[5] Tutorialspoint. (2020). Redis – Overview. Retrieved from: https://www.tutorialspoint.com/redis/redis\_overview.htm [Accessed 07 Dec. 2020].

[6] Wikipedia. (2020). Object – relational mapping. Retrieved from: https://en.wikipedia.org/wiki/Object-relational\_mapping [Accessed 07 Dec. 2020].

[7] TypeORM. (2020). TypeORM - Amazing ORM for TypeScript and JavaScript (ES7, ES6, ES5). Supports MySQL, PostgreSQL, MariaDB, SQLite, MS SQL Server, Oracle, WebSQL databases. Works in NodeJS, Browser, Ionic, Cordova and Electron platforms. Retrieved from: https://typeorm.io [Accessed 07 Dec. 2020].

[8] Tutorialspoint. (2020). Node.js - Introduction. Retrieved from: https://www.tutorialspoint.com/nodejs/nodejs\_introduction.htm [Accessed 08 Dec. 2020].

[9] Wikipedia. (2020). NodeJS – History section. Retrieved from: https://en.wikipedia.org/wiki/Node.js#History [Accessed 08 Dec. 2020].

[10] Tutorialspoint. (2020). TypeScript - Overview. Retrieved from: https://www.tutorialspoint.com/typescript/typescript\_overview.htm [Accessed 08 Dec. 2020].

[11] Wikipedia. (2020). TypeScript – History section. Retrieved from: https://en.wikipedia.org/wiki/TypeScript#History [Accessed 08 Dec. 2020].

[12] NestJS. (2020). Documentation | NestJS - A progressive Node.js framework. Retrieved from: https://docs.nestjs.com [Accessed 08 Dec. 2020].

[13] Tutorialspoint. (2020). ReactJS - Overview. Retrieved from: https://www.tutorialspoint.com/reactjs/reactjs\_overview.htm [Accessed 08 Dec. 2020].

[14] Wikipedia. (2020). React – History section. Retrieved from: https://en.wikipedia.org/wiki/React\_(web\_framework)#History [Accessed 08 Dec. 2020].

[15] NextJS. (2020). Create a Next.js App | Learn Next.js. Retrieved from: https://nextjs.org/learn/basics/create-nextjs-app [Accessed 10 Dec. 2020].

[16] Flavio Copes. (2019). The Next.js Handbook. Retrieved from: https://www.freecodecamp.org/news/the-next-js-handbook [Accessed 10 Dec. 2020].

[17] Ant Design. (2020). Ant Design of React. Retrieved from: https://ant.design/docs/react/introduce [Accessed 10 Dec. 2020].

[18] Wikipedia. (2020). Docker (software). Retrieved from: https://en.wikipedia.org/wiki/Docker\_(software) [Accessed 12 Dec. 2020].

[19] CodeShip. (2020). Continuous Integration Essentials. Retrieved from: https://codeship.com/continuous-integration-essentials [Accessed 12 Dec. 2020].

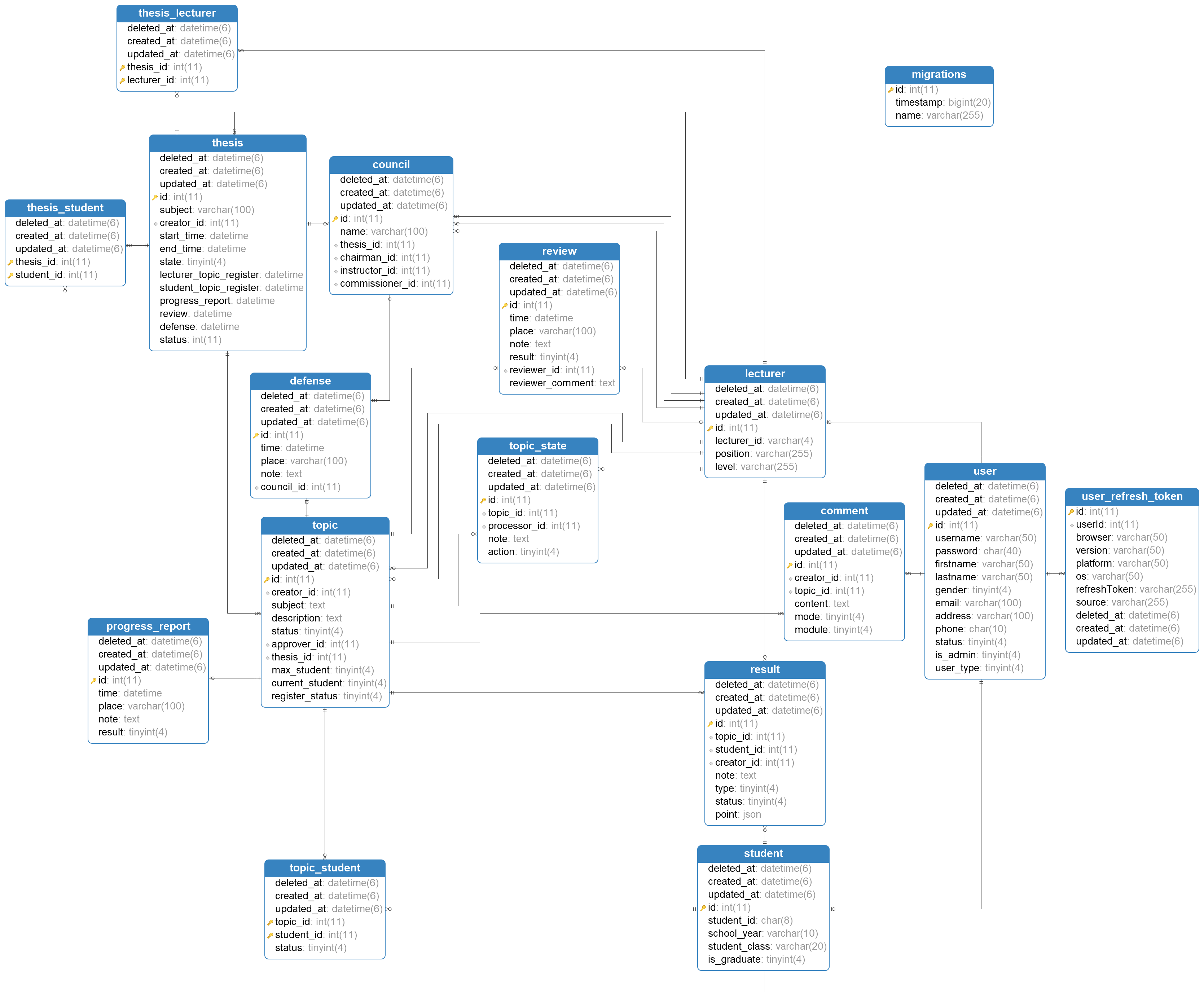
[20] GitHub. (2020). Product – GitHub Actions. Retrieved from: https://docs.github.com/en/free-pro-team@latest/actions [Accessed 12 Dec. 2020].

[21] AWS. (2020). Amazon S3. Retrieved from: https://aws.amazon.com/s3 [Accessed 12 Dec. 2020].

[22] Heroku. (2020). The Heroku Platform. Retrieved from: [Accessed 12 Dec. 2020].

[23] Postman. (2020). Postman | The Collaboration Platform for API Development. Retrieved from: https://www.postman.com [Accessed 12 Dec. 2020].

# APPENDIX 1: DATABASE DIAGRAM



# APPENDIX 2: SCREEN FLOW

