

CAMT Exam Dashboard with Tesseract

Software Design Document

By

Lingyu Kong 612115506

Jiajun Tao 612115503

BACHELOR OF SOFTWARE ENGINEERING PROGRAM

COLLEGE OF ARTS, MEDIA AND

TECHNOLOGY CHIANG MAI UNIVERSITY

Advisor

Pree Thiengburanathum,PhD

CAMT Exam Dashboard with Tesseract

Lingyu Kong 612115506

Jiajun Tao 612115503

THIS REPORT HAS BEEN APPROVED TO BE A PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE SOFTWARE ENGINEERING PROGRAM: COLLEGE OF ARTS MEDIA AND TECHNOLOGY

…………………………………………………. ADVISOR

Dr. PREE THIENGBURANATHUM

…………………………………………………...MEMBER

Lingyu Kong

…………………………………………………...MEMBER

Jiajun Tao

**Document History**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Version** | **History** | **Status** | **Date** | **View-able** | **Editable** | **Respon-sible** |
| SDD\_Version.1.docx | **Add**   * Chapter 1 * Chapter 2 | Draft | 1  June , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.2.docx | **Add**   * Chapter 3 | Draft | 4  June , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.3.docx | **Add**   * Chapter 1 * Chapter 4 | Draft | 7  June , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.4.docx | **Update**   * Sequence Diagram * Class Diagram Description | Draft | 27  June , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.5.docx | **Update**   * UI Design * Correct Spelling * Correct Grammar mistakes | Draft | 28  June , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.6.docx | **Add**   * Database Design * ER Diagram | Draft | 1  July , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.7.docx | **Update**   * Sequence Diagram * Class Diagram * Class Diagram Description | Draft | 30  July , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.8.docx | **Update**   * Adjustment document format | Release | 10  Aug , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.9.docx | **Update**   * Chapter 2 * Database design | Release | 10  Sep , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.10.docx | **Add**   * Chapter 1 * Chapter 2 | Draft | 1  Sep , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.11.docx | **Add**   * Class diagram * Database design | Draft | 4  Sep , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.12.docx | **Add**   * Sequence diagram   Chapter 4 | Draft | 10  Sep , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.13.docx | **Update**   * Sequence diagram * Class diagram   UI design | Draft | 15  Sep , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.14.docx | **Update**   * Adjustment document format | Release | 16  Sep , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.15.docx | **Update**   * Chapter 3 * Edit exam info Sequence diagram | Release | 26  Sep , 2021 | Advisor  LK  JT | LK  JT | LK  JT |
| SDD\_Version.16.docx | **Update**   * 3.1.1 Feature 1: Authentication System * 3.1.2 Feature 2: Table and cell detection and text exstraction and parsing System * 3.1.3 Feature 3: Dashboard System * 3.1.4 Feature 4: Notification System * Chapter 4 * 3.3 sequence diagram | Release | 10  Oct , 2021 | Advisor  LK  JT | LK  JT | LK  JT |

**\*LK = Lingyu Kong**

**\*JT = Jiajun Tao**

**\*** **Advisor = Pree Thiengburanathum,PhD**

Table of Contents

Chapter One | Introduction 6

1.1 Purpose 6

1.2 Project Overview………………………………………………………………………………………………. 6

1.2.1 Project Scope 6

1.2.2 Document Scope 6

1.3 Acronyms and Definition…………………………………………………………………………………………………...7

1.3.1 Acronyms 7

1.3.2 Definition 7

Chapter Two | System Architecture 9

Chapter Three | Detailed Design 11

3.1 Class Diagram………………………………………………………………………………………………. 11

3.1.1 Feature 1: Authentication System 11

3.1.2 Feature 2: Table and cell detection and text exstraction and parsing System 19

3.1.3 Feature 3: Dashboard System 37

3.1.4 Feature 4: Notification System 47

3.2 Database Design ………………………………………………………………………………………………. 55

3.3 Sequence Diagram………………………………………………………………………………………………. 59

Chapter Four | User Interface Design 67

4.1 Login page………………………………………………………………………………………………. 67

4.2 Logout page………………………………………………………………………………………………. 68

4.3 ParseFile page………………………………………………………………………………………………. 69

4.4 Dashboard page………………………………………………………………………………………………. 71

4.5 ExamInfoList page………………………………………………………………………………………………. 72

4.5 Edit ExamInfo page………………………………………………………………………………………………. 73

4.7 Send Email page………………………………………………………………………………………………. 74

Chapter One | Introduction

* 1. Purpose

The purpose of the Software Design Document is to provide a description of the design of a system fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to build. The Software Design Document provides information necessary to provide a description of the details for the software and system to be built.

1.2 Project Overview

The CAMT exam dashboard with Tesseract is a web-based application designed to help lecturers effectively access exam information and assist staff in managing exam information. The system is built using Vue.js, SpingMVC and Mybaties, and is integrated with CMU API to facilitate lecturers to log in to the system. In addition, use SpirePDF, OPENCV and TencentOCR engines to parse the PDF exam schedule which uploaded by the lecturer or admin into an Excel table for the lecturer or admin to download, and store the data in the database at the same time. We also use the Echarts dashboard framework to generate exam overviews, professional classifications, and exam time distribution for lecturers and admin. Finally, we also use the Java Mail API to send modify and delete the exam schedule notification to the lecturers via CMU mail box.

1.2.1 Project Scope

The CAMT exam dashboard with Tesseract has four features, which are including these scopes:

● Feature#1: Authentication System

The administrator uses the administrator account to log in to the system

The lecturer uses the CMU account to log in to the system.

● Feature#2: Table and cell detection and text extraction and parsing system

The system detects and parses the manual input PDF to generates EXCEL and data frame.

● Feature#3: Dashboard System

The system generates a dashboard based on the data frame that generated by the detection and analysis.

● Feature#4: Notification System

The administrator can modify the information in the database and send exam notifications for the protector.

1.2.2 Document Scope

This document includes System architecture, Class diagram, Class diagram description, User Interface design, Function Group, Method description, and Sequence diagram.

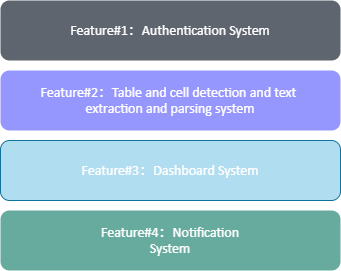
* 1. Acronyms and Definition
     1. Acronyms

|  |  |
| --- | --- |
| SD | Sequence Diagram |
| CD | Class Diagram |
| UI | User Interface |
| CDD | Class Diagram Description |
| MD | Method Description |

* + 1. Definition

|  |  |
| --- | --- |
| Name | Definition |
| IEEE | Institute for Electrical and Electronics Engineers. Biggest global interest group for engineers of different branches and computer scientists. [IEEE90] |
| Interface | 1) A shared boundary across which information is passed.  2) A hardware or software component that connects two or more other components for the purpose of passing information from one to the other.  3) To connect two or more components for the purpose of passing information from one to the other.  4) To serve as a connecting or connected component as in (2). [IEEE90] |
| Data | 1) A representation of facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means.  2) Something used as a synonym for documentation. [IEEE90] |
| Architecture | The organizational structure of a system or component. [IEEE90] |

Chapter Two | System Architecture



The architecture of CAMT Exam Dashboard with Tesseract shown in figure 1. It consists of five systems.

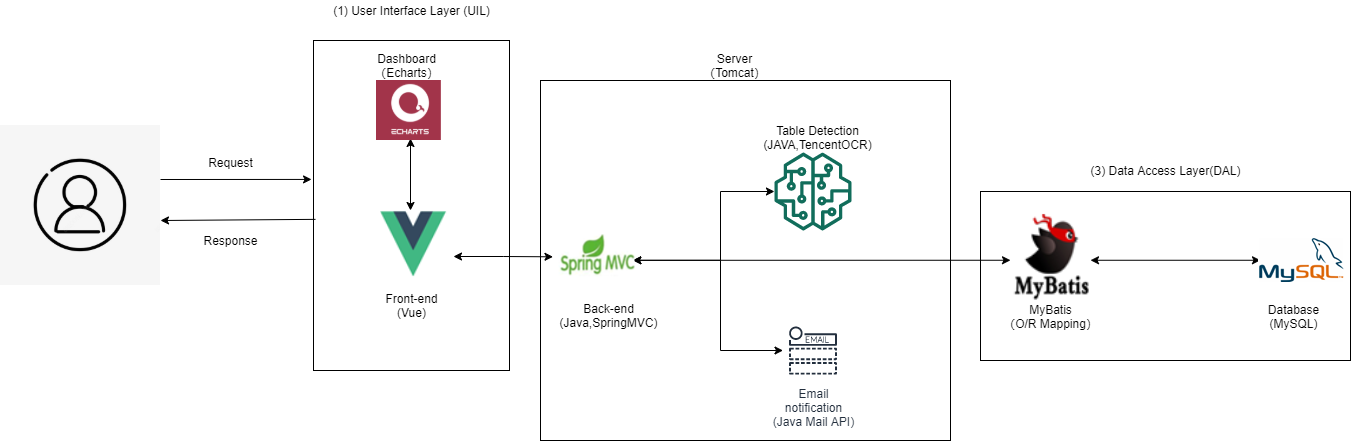
- The authentication system provides users login feature. The lecturer can log in with the CMU IT account, and the administrator can log in with the administrator account.

- The table and cell detection and text extraction and parsing system is the system for users to parse the CAMT exam timetable.

- The dashboard system is a system for users to view the exam summary.

- The notification system is a system for administrators to modify exam data and send exam notifications to protectors.

Figure 2 System Architecture

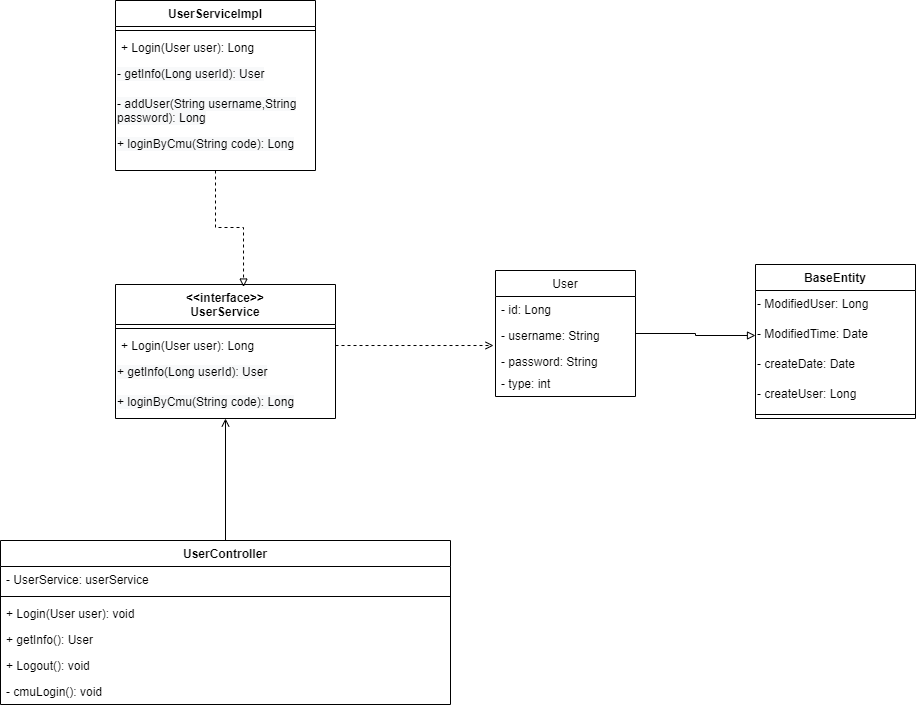


In figure 2, users send requests or get responses from the system. Frontend shows the user interface to users and receives requests from users and does responses by using Vue.js. The server connects the frontend and backend by using the Tomcat. Backend stores the data of the system by using the MySQL database.

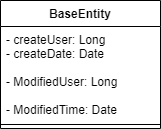
Chapter Three | Detailed Design

3.1 Class Diagram

3.1.1 Feature 1: Authentication System

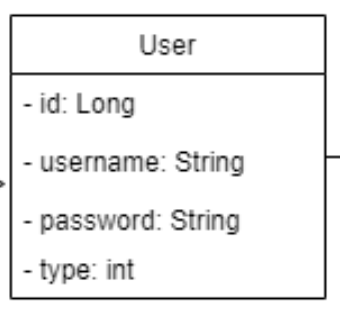


**CDD-1: BaseEntity**

****

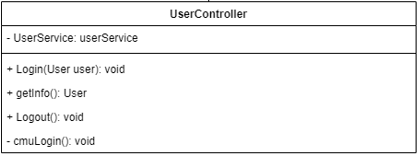
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BaseEntity** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | createUser | | Data create user | | Long |
| 2 | createDate | | Data create time | | Date |
| 3 | ModifiedUser | | Data modify user | | Long |
| 4 | ModifiedTime | | Data modify time | | Date |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| - | - |  | | - | - |

**CDD-2: User**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | id | | the user ID is in the database, starting with recount 1. | | Long |
| 2 | username | | Username (string of length between 4 and 32) | | String |
| 3 | password | | User password (string of length at least 6 and includes a digit number) | | String |
| 4 | type | | User type(1:admin, 0: user) | | Int |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| - | - |  | | - | - |

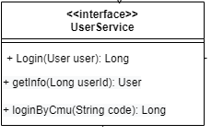
**CDD-3: UserController**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UserController** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | UserService | | Call the Userservice layer | | UserService |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD1 | Login(User user): void | User object | | This function used to receive user login requests and validate login information. | Return login response in JSON format. |
| MD2 | getInfo(): User | - | | This function gets user information from database. | Return the user info in JSON format. |
| MD3 | Logout(): void | user | | This function used to receive user logout requests from frontend, and remove token. | Return the user token has been remove. |
| MD4 | cmuLogin():void | - | | This function used to receive cmu login requests and get user basic information from ITSC. | Return login response in JSON format. |

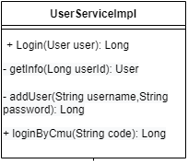
**CDD-4: <<interface>>**

**UserService**

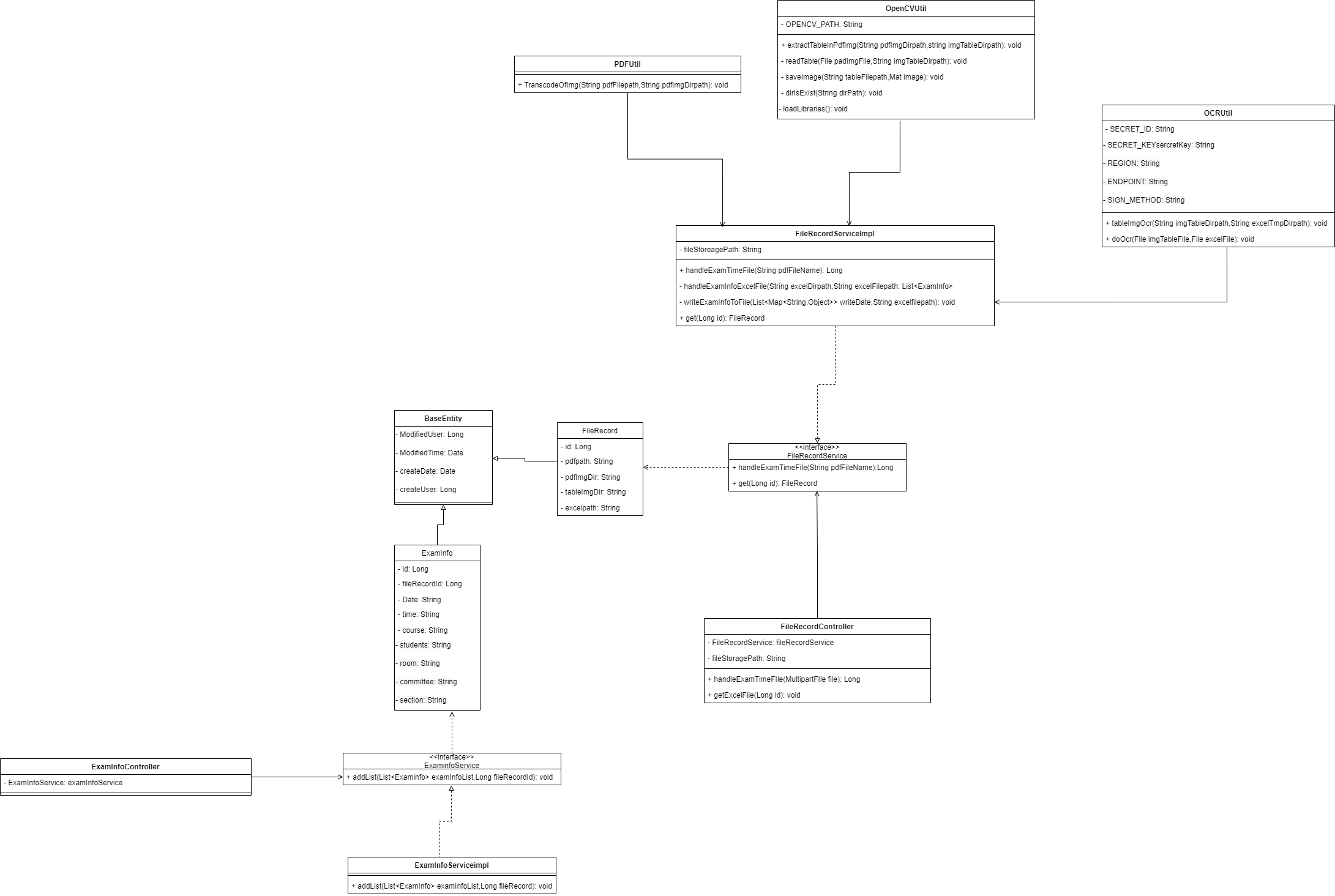


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **<<interface>> UserService** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| - | - | | - | | - |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD5 | Login(User user):Long | User object | | Interface, used to call the Login() method in UserserviceImpl. | - |
| MD6 | getInfo(Long userId): User | User id | | Interface, used to call the getInfo(Long userId) method in UserserviceImpl. | - |
| MD7 | loginByCmu(String code):Long | Authorization code | | Interface, used to call the LoginByCmu() method in UserserviceImpl. |  |

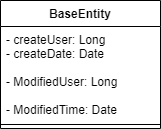
**CDD-5: UserServicelmpl**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UserServicelmpl** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| - | - | | - | | - |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD8 | Login(User user):Long | User object | | This function validate the username and password from the database. | 1. Return User Id  2. Return an error message” username or password not match  ” If the admin input the error username or password |
| MD9 | getInfo(Long userId): User | User id | | This function gets user information by user ID. | 1. Return User information.  2. Return an error message” user not found, id: xxx” If cannot find user form database |
| MD10 | addUser(String username, String password):Long | Username,  password | | This function use to add new user data in the database. | 1. Return User Id  2. Return an error message” add new user data failed, please contact admin” If insert data in database failed. |
| MD11 | LoginByCmu(String code):Long | Authorization code | | This function use to get access code by using Authorization code, request user basic information by using access code and add user data in the database. | 1. Return User Id  2. Return an error message ” login failed, please contact admin” If response is null.  3. Return an error message ” login failed, please contact admin” If access token is empty.  4. Return an error message ” login failed, please contact admin” If user information is null. |

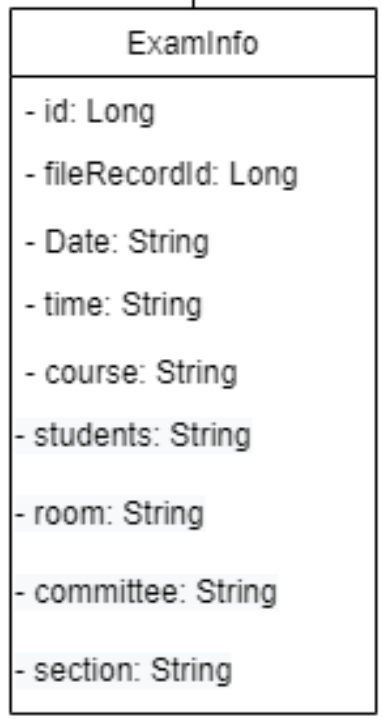
****3.1.2 Feature 2: Table and cell detection and text extraction and parsing system.

**CDD-1: BaseEntity**

****

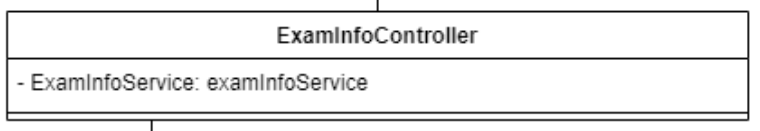
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BaseEntity** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | createUser | | Data create user | | Long |
| 2 | createDate | | Data create time | | Date |
| 3 | ModifiedUser | | Data modify user | | Long |
| 4 | ModifiedTime | | Data modify time | | Date |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| - | - |  | | - | - |

**CDD-6: ExamInfo**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ExamInfo** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | Id | | The exam information ID is in the database, starting with recount 1. | | Long |
| 2 | fileRecordId | | The corresponding file record id | | Long |
| 3 | date | | a brief text of date | | String |
| 4 | time | | a brief text of time | | String |
| 5 | course | | a brief text of course | | String |
| 6 | students | | a brief text of students | | String |
| 7 | room | | a brief text of room | | String |
| 8 | committee | | a brief text of committee | | String |
| 9 | section | | a brief text of section | | String |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| - | - |  | |  | - |

**CDD-7: ExamInfoController**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ExamInfoController** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | examInfoService | | Call the ExamInfoService layer | | ExamInfoService |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| - | - |  | |  | - |

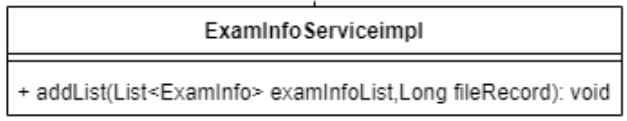
**CDD-8: <<interface>>**

**ExaminfoService**



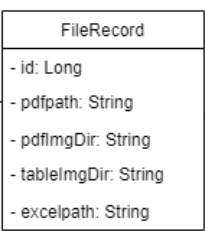
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **<<interface>> ExaminfoService** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| - | - | | - | | - |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD12 | addList(List<Examinfo> examInfoList,Long fileRecordId) | examInfoList,fileRecordId | | Interface,  To call addList method in ExamInfoServiceImpl | - |

**CDD-9: ExamInfoServiceimpl**



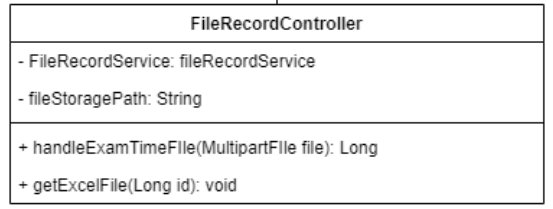
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ExamInfoServiceimpl** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| - | - | | - | | - |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD13 | addList(List<ExamInfo> examInfoList,Long fileRecord): void | examInfoList,  fireRecord | | This function insert exam information to the database base on fireReocrd Id. | 1. Return an error message" insert exam Info data failed，please contact admin" If insert exam Info data failed. |

**CDD-10: FileRecord**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FileRecord** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | Id | | the file record ID is in the database, starting with recount 1. | | Long |
| 2 | pdfpath | | a string of store the pdf file path | | String |
| 3 | pdfImgDir | | a string of store the image path | | String |
| 4 | tableImgDir | | a string of store the table image path | | String |
| 5 | excelpath | | a string of store the excel path | | String |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| - | - | - | | - | - |

**CDD-11: FileRecordController**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FileRecordController** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | fileRecordService: | | Call the FileRecordService layer | | FileRecordService |
| 2 | fileStoragePath | | Download excel file path | | String |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  14 | handleExamTimeFIle(MultipartFIle file): Long | file | | This function used to receive user parse file requests and verify that the user uploads the file format and then parse the file. | 1. Return response in JSON format.  2.Return file record id.  3. Return an error message:” file is empty” if user upload empty pdf file.  4. Return an error message" file size can’t not exceed" If uploaded file size over max size  5. Return an error message" file type does not meet the requirements, only is pdf" If uploaded file is not a pdf  6. Return an error message" original file may have been deleted, and the original file cannot be accessed " If original file is missing.  7. Return an error message" read and write error occurred while processing the file!" If read or write file fails |
| MD  15 | getExcelFile(Long id): void | File record id | | This function used to receive user download excel file requests and gets a parsed excel file. | 1. Return excel file .  2. Return an error message" file download failed，please contact admin " If file download fails.  3. Return an error message" no such file, file name + $filename" If can not find the file. |

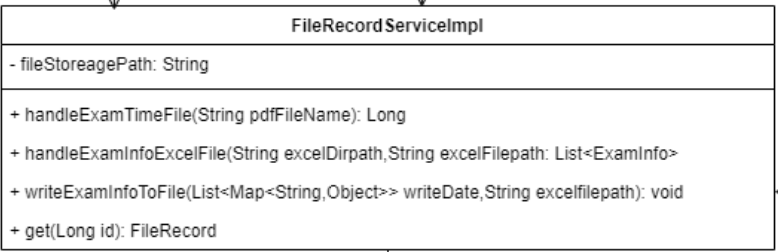
**CDD-12: <<interface>>**

**FileRecordService**



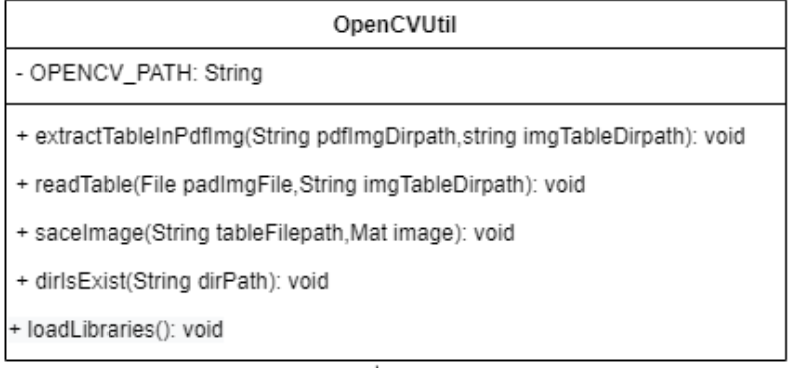
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **<<interface>> FileRecordService** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| - | - | | - | | - |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  16 | handleExamTimeFile(String pdfFileName):Long | pdfFile | | Interface, used to call the handleExamTimeFIle() method in FileRecordserviceImpl. | - |
| MD  17 | get(Long id): FileRecord | File record id | | Interface, used to call the get() method in FileRecordserviceImpl. | - |

**CDD-13: FileRecordServiceImpl**



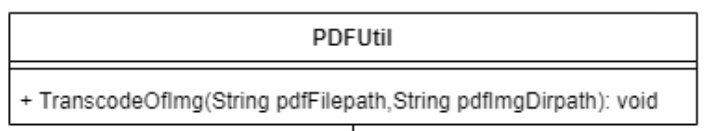
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FileRecordServiceImpl** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | fileStoreagePath | | Download excel file path | | String |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  18 | handleExamTimeFile(String pdfFileName): Long | pdfFileName | | This function parsing uploaded pdf files into Excel file | 1. Return file record id  2. Return an error message" update file record data failed，please contact admin" If update file record data failed.  3. Return an error message" insert fileRecord data failed, please contact admin" If create new file record failed. |
| MD  19 | handleExamInfoExcelFile(String excelDirPath,String excelFilePath):List<ExamInfo> | ExcelDiepath,excelFilePath | | This function merges the parsed Excel and save it to the path | 1.Return examInfoList |
| MD  20 | writeExamInfoToFile(List<Map<String,Object>> writeData,String excelfilepath): void | WriteData,  excelfilepath | | This function will create the Excel file and write the examinfo data into the excel file. | - |
| MD  21 | get(Long id): FileRecord | - | | This function gets a file record data from the database. | 1. Return file record data. |

**CDD-14: OpenCVUtil**



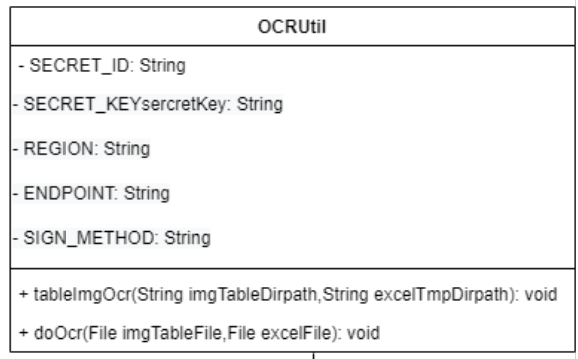
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **OpenCVUtil** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | OPENCV\_PATH | | OPENCV library path | | String |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  22 | extractTableInPdfImg(String pdfImgDirpath,string imgTableDirpath): void | pdfImgDirPath,imgTableDirpath | | This function check if there are files under the path which can be extracted a table from a PDF image and then extract table structure. | 1. Return an error message” no such file to extract table，please contact admin" If cannot find any files under the path |
| MD  23 | readTable(File padImgFile,String imgTableDirPath): void | pdfImgFile,  imgTableDirPath | | This function extract table from PDF image | - |
| MD  24 | saveImage(String tableFilepath,Mat table): void | tableFilePath,  table | | This function checks whether the directory exists and if exist saves the table image file | - |
| MD  25 | dirIsExist(String dirPath): void | dirPath | | This function checks whether the directory exists | - |
| MD  26 | loadLibraries(): void | - | | This function for load libraries | 1. Return an error message "Failed to load OpenCV native library，please contact admin" If load libraries fails |

**CDD-15: PDFUtil**



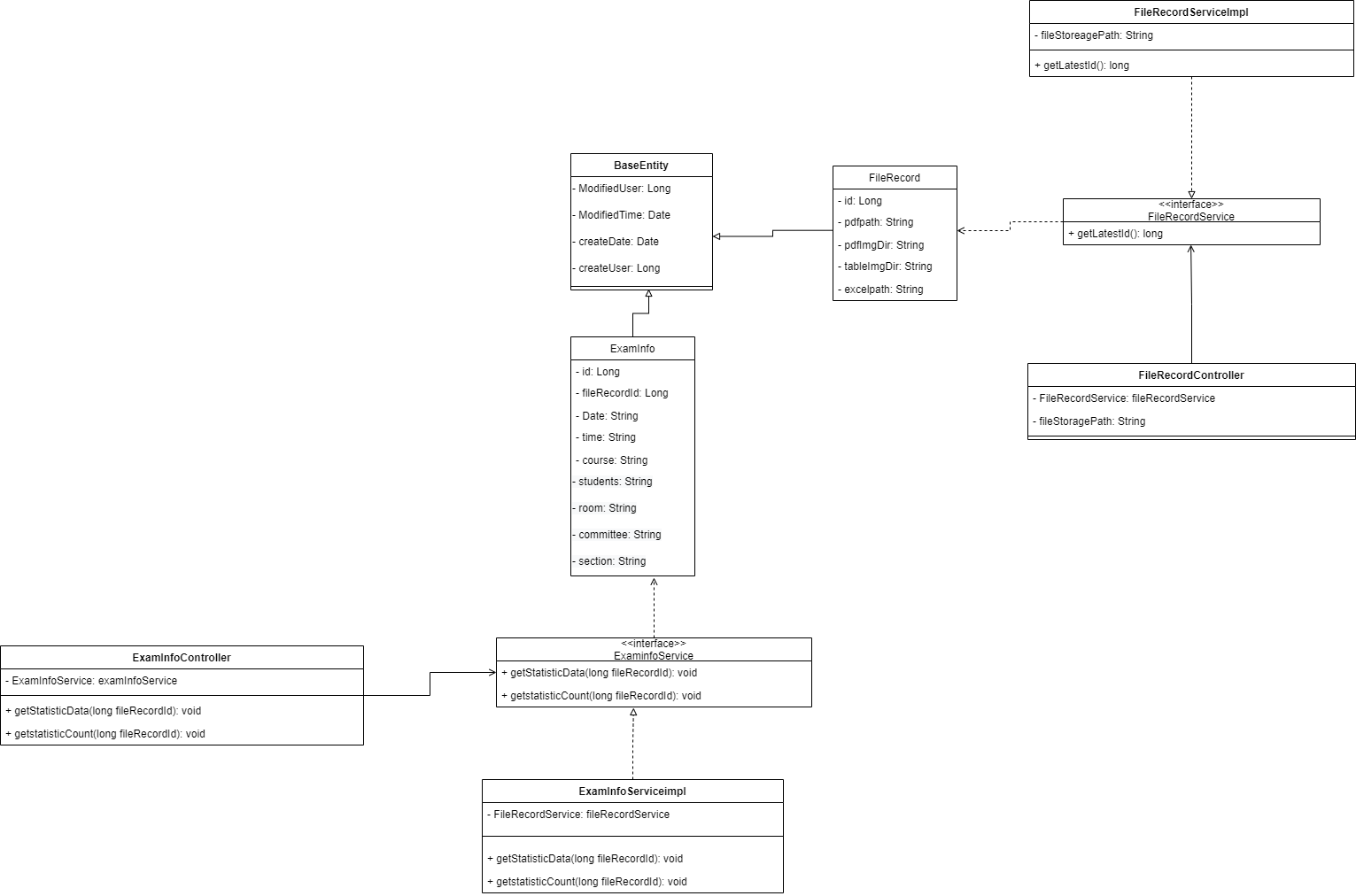
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PDFUtil** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| - | - | | - | | - |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  27 | TranscodeOfImg(String pdfFilePath,String pdfImgDirPath): void | pdfFilePath,  pdfImgDirPath | | This function transcodes the PDF file into image files. | 1. Return an error message "pdf transcode image fail, please contact admin" If pdf transcode image fail |

**CDD-16: OCRUtil**

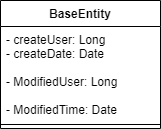


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **OCRUtil** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | SECRET\_ID | | Id for access TencentOCR api | | String |
| 2 | SECRET\_KEYsercretKey | | Keysercretkey for access TencentOCR api | | String |
| 3 | REGION | | Region for access TencentOCR api | | String |
| 4 | ENDPOINT | | Endpoint for using TencentOCR api | | String |
| 5 | SIGN\_METHOD | | Which service of Tencent OCR to use | | String |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  28 | tableImgOcr(String imgTableDirpath,String excelTmpDirpath): void | imgTableDirpath,  excelTmpDirpath | | This function verifies that there is a file in the directory that can be parsed, and then call doOcr method to parse the table image file. | 1. Return an error message" no such file to extract table，please contact admin" If no file to extract |
| MD  29 | doOcr(File imgTableFile,File excelFile): void | imgTableFile,  excelFile | | This function call Tencent OCR recognition engine to parse the table image file. | 1. Return an error message" table image ocr fail，please contact admin" If call Tencent OCR engine fails |

3.1.3 Feature 3: Dashboard System

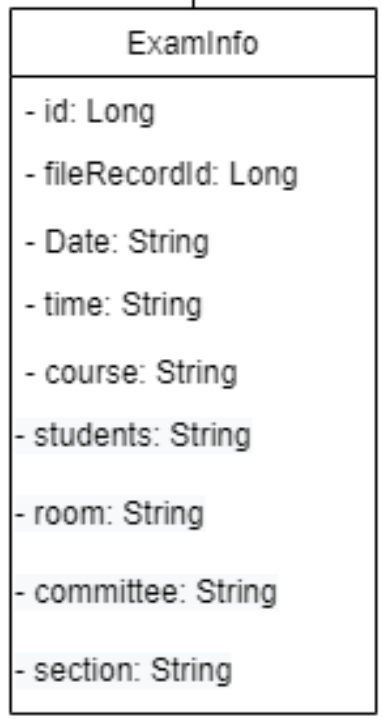


**CDD-1: BaseEntity**

****

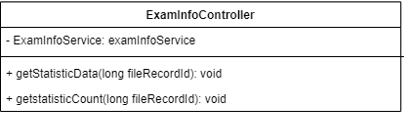
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BaseEntity** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | createUser | | Data create user | | Long |
| 2 | createDate | | Data create time | | Date |
| 3 | ModifiedUser | | Data modify user | | Long |
| 4 | ModifiedTime | | Data modify time | | Date |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| - | - |  | | - | - |

**CDD-6: ExamInfo**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ExamInfo** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | Id | | The exam information ID is in the database, starting with recount 1. | | Long |
| 2 | fileRecordId | | The corresponding file record id | | Long |
| 3 | date | | a brief text of date | | String |
| 4 | time | | a brief text of time | | String |
| 5 | course | | a brief text of course | | String |
| 6 | students | | a brief text of students | | String |
| 7 | room | | a brief text of room | | String |
| 8 | committee | | a brief text of committee | | String |
| 9 | section | | a brief text of section | | String |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| - | - |  | |  | - |

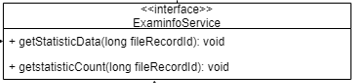
**CDD-17: ExamInfoController**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ExamInfoController** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | examInfoService | | Call the ExamInfoService layer | | ExamInfoService |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  30 | getStatisticData() | fileRecordId | | This function use to get exam information: groupbydate, groupbycourse, groupbyprofessional from database. | Return exam data groupbydate, groupbycourse, groupbyprofessional in JSON format. |
| MD  31 | getStatisticCount() | fileRecordId | | This function use to get exam information: countbycourse, countbyteacher, countbysection from database. | Return exam data countbycourse, countbyteacher, countbysection in JSON format. |

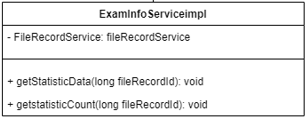
**CDD-18: <<interface>>**

**ExaminfoService**



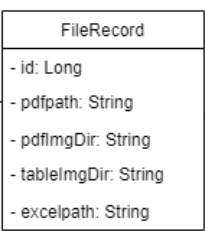
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **<<interface>> ExaminfoService** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| - | - | | - | | - |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  32 | getStatisticData() | fileRecordId | | Interface, used to call the getStatisticData() method in ExamInfoserviceImpl. | - |
| MD  33 | getStatisticCount() | fileRecordId | | Interface, used to call the getStatisticCount() method in ExamInfoserviceImpl. | - |

**CDD-19: ExamInfoServiceimpl**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ExamInfoServiceimpl** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | fileRecordService | | fileRecordService | | FileRecordService |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  34 | getStatisticData() | fileRecordId | | This function use to get exam information: groupbydate, groupbycourse, groupbyprofessional from database. | Return select data groupByDate, groupByCourse, groupByProfession from database. |
| MD  35 | getStatisticCount() | fileRecordId | | This function use to get exam information: countbycourse, countbyteacher, countbysection from database. | Return select data countbycourse, countbyteacher, countbysection from database. |

**CDD-10: FileRecord**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FileRecord** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | Id | | the file record ID is in the database, starting with recount 1. | | Long |
| 2 | pdfpath | | a string of store the pdf file path | | String |
| 3 | pdfImgDir | | a string of store the image path | | String |
| 4 | tableImgDir | | a string of store the table image path | | String |
| 5 | excelpath | | a string of store the excel path | | String |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| - | - | - | | - | - |

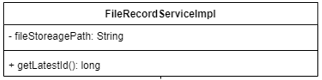
**CDD-20: <<interface>>**

**FileRecordService**

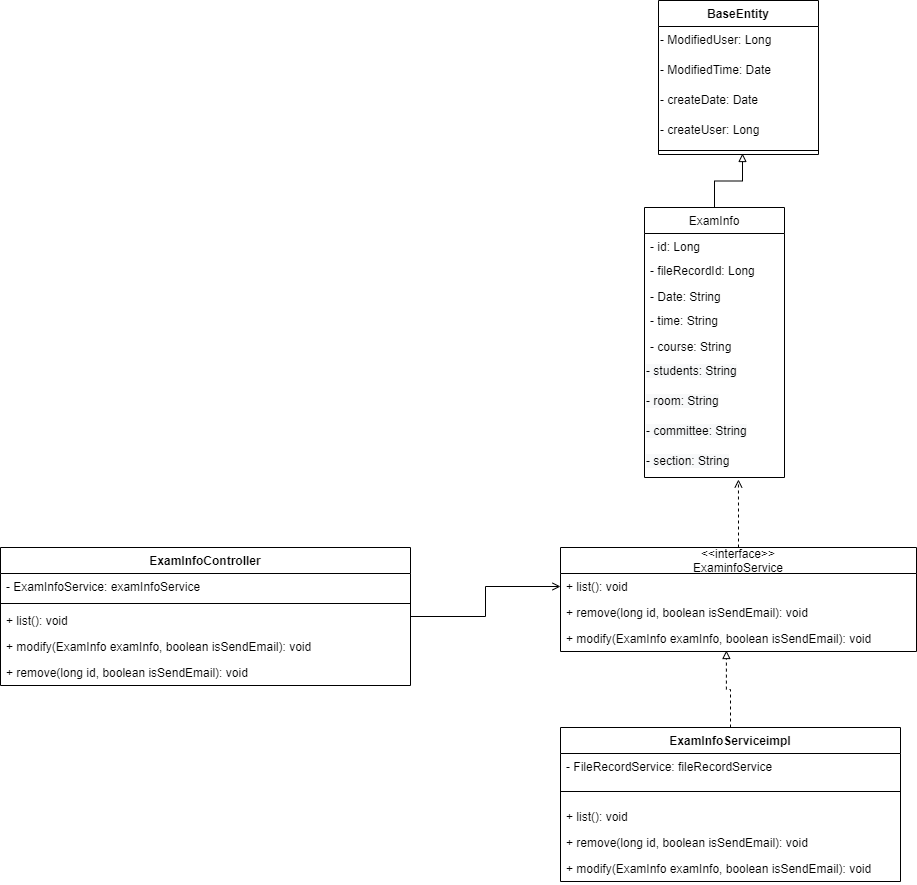


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **<<interface>> FileRecordService** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| - | - | | - | | - |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  36 | getLastestId() |  | | Interface, used to call the getLastestId() method in FileRecordserviceImpl. | - |

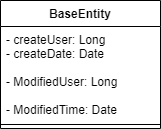
**CDD-21: FileRecordServiceImpl**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FileRecordServiceImpl** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | fileStoreagePath | | Download excel file path | | String |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  37 | getLastestId() | - | | This function get lastest fileRecord Id from the database. | 1. Return FileRecord id.  2. Return “no fileRecord” if no fileRecord in the database. |

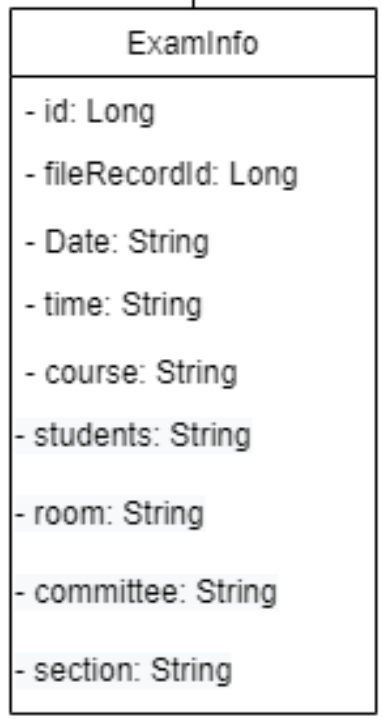
3.1.4 Notification System

**CDD-1: BaseEntity**

****

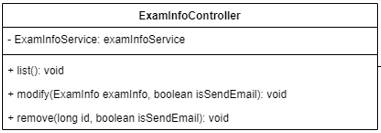
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BaseEntity** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | createUser | | Data create user | | Long |
| 2 | createDate | | Data create time | | Date |
| 3 | ModifiedUser | | Data modify user | | Long |
| 4 | ModifiedTime | | Data modify time | | Date |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| - | - |  | | - | - |

**CDD-6: ExamInfo**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ExamInfo** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | Id | | The exam information ID is in the database, starting with recount 1. | | Long |
| 2 | fileRecordId | | The corresponding file record id | | Long |
| 3 | date | | a brief text of date | | String |
| 4 | time | | a brief text of time | | String |
| 5 | course | | a brief text of course | | String |
| 6 | students | | a brief text of students | | String |
| 7 | room | | a brief text of room | | String |
| 8 | committee | | a brief text of committee | | String |
| 9 | section | | a brief text of section | | String |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| - | - |  | |  | - |

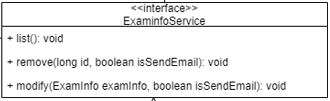
**CDD-22: ExamInfoController**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ExamInfoController** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | examInfoService | | Call the ExamInfoService layer | | ExamInfoService |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  38 | list() | - | | This function get exam information from the database and show the list in the page. | Return list response in JSON format. |
| MD  39 | modify() | examInfo, isSendEmail | | This function use to edit exam information to the database and send email notification to committee. | Return edit response in JSON format and send email notification. |
| MD  40 | remove() | Id,  isSendEmail | | This function use to delete exam information to the database and send email notification to committee. | Return delete response in JSON format and send email notification. |

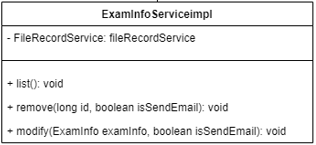
**CDD-23: <<interface>>**

**ExaminfoService**



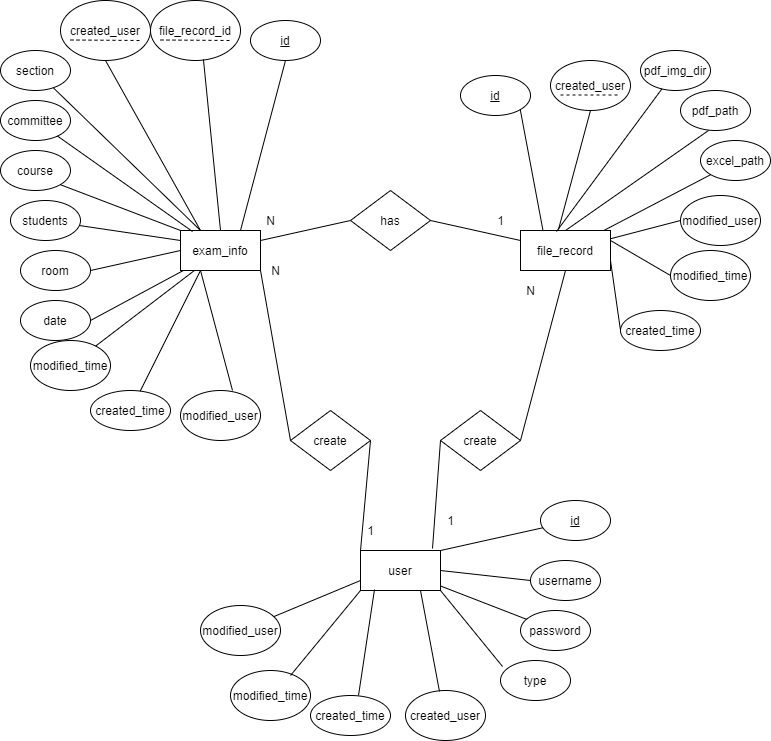
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **<<interface>> ExaminfoService** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| - | - | | - | | - |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  41 | list() | - | | Interface, used to call the list() method in ExamInfoserviceImpl. | - |
| MD  42 | modify() | examInfo, isSendEmail | | Interface, used to call the modify() method in ExamInfoserviceImpl. | - |
| MD  43 | remove() | Id,  isSendEmail | | Interface, used to call the remove() method in ExamInfoserviceImpl. | - |

**CDD-24: ExamInfoServiceimpl**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ExamInfoServiceimpl** | | | | | |
| Attributes | | | | | |
| ID | Name | | Description | | Type |
| 1 | fileRecordService | | fileRecordService | | FileRecordService |
| Methods | | | | | |
| ID | Name | Parameters | | Description | Return |
| MD  44 | list() | - | | This function get exam information from the database and show the list in the page. | Return exam information list. |
| MD  45 | modify() | examInfo, isSendEmail | | This function use to edit exam information to the database and call sendEmail() method send Email notification to the committee. | 1.Return an error message “update examInfo data failed, please contact admin” If update data in database failed. |
| MD  46 | remove() | Id,  isSendEmail | | This function use to delete exam information to the database and call sendEmail() method send Email notification to the committee. | Return an error message “delete examInfo data failed, please contact admin” If update data in database failed. |
| MD  46 | sendEmail() | Title,  examInfo | | This function use to send exam notification. | 1. Return an error message “recipient email is invalid form: ($email)” If email is blank or invalid. |

3.2 Database Design



3.2.1 Data Description

user table

Description: Used to store user information

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO. | Name | Description | Type | Size | Key |
| 1 | id | This attribute is the id of the user | Int | 11 | PK |
| 2 | username | This attribute is the username of the user | varchar | 20 | - |
| 3 | password | This attribute is the password of the user | varchar | 20 | - |
| 4 | type | This attribute is the type of the user | TINYINT | 1 | - |
| 5 | created\_time | This attribute is the created time of the user | datetime | - | - |
| 6 | modified\_time | This attribute is the modified time of the user | datetime | - | - |
| 7 | created\_user | This attribute is the created user of the user | Int | 11 | - |
| 8 | modified\_user | This attribute is the modified user of the user | int | 11 | - |

Exam\_info table

Description: Used to store exam information

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO. | Name | Description | Type | Size | Key |
| 1 | id | This attribute is the id of the exam\_info | Int | 11 | PK |
| 2 | file\_record\_id | This attribute is the id of the file\_record\_id table | Int | 11 | FK |
| 3 | date | This attribute is the exam date of the exam\_info | varchar | 64 | - |
| 4 | time | This attribute is the exam time of the exam\_info | Varchar | 32 | - |
| 5 | course | This attribute is the course of the exam\_info | Varchar | 16 | - |
| 6 | section | This attribute is the section of the exam\_info | Varchar | 16 |  |
| 7 | students | This attribute is the student of the exam\_info | Varchar | 16 | - |
| 8 | room | This attribute is the room of the exam\_info | Varchar | 32 | - |
| 9 | committee | This attribute is the committee of the exam\_info | Varchar | 32 | - |
| 10 | created\_time | This attribute is the created time of the exam\_info | Datetime | - | - |
| 11 | modified\_time | This attribute is the modified time of the exam\_info | Datetime | - | - |
| 12 | created\_user | This attribute is the created user of the exam\_info | Int | 11 | FK |
| 13 | modified\_user | This attribute is the modified user of the exam\_info | int | 11 | - |

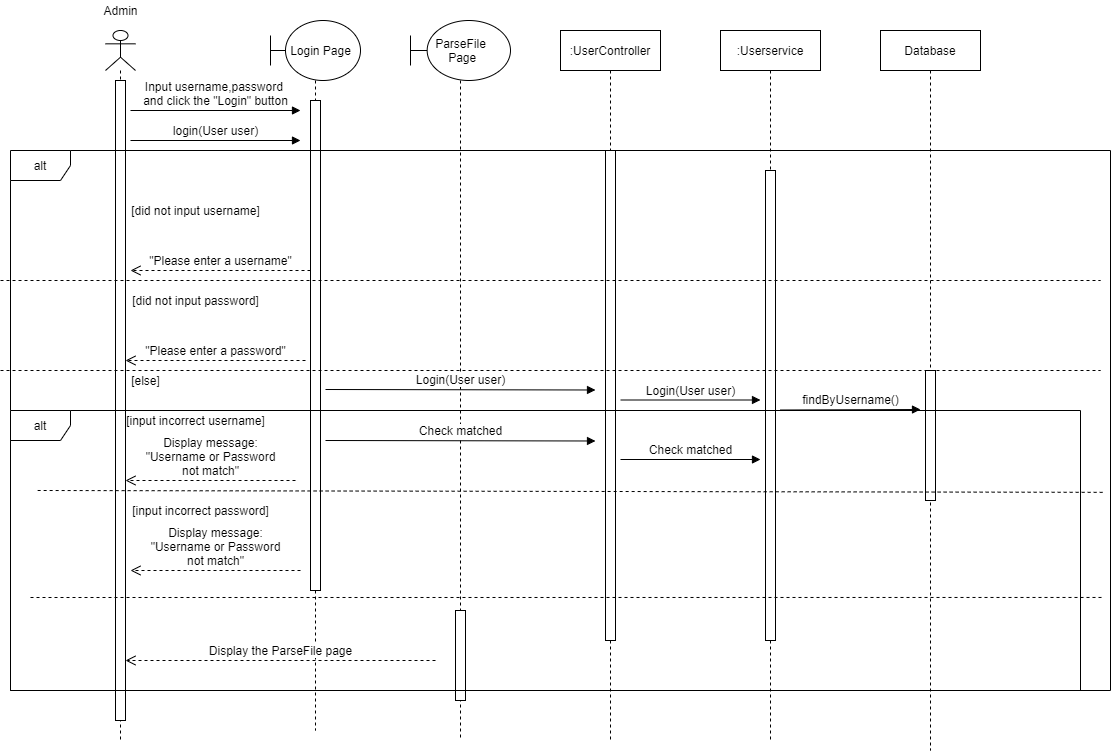
File\_record table

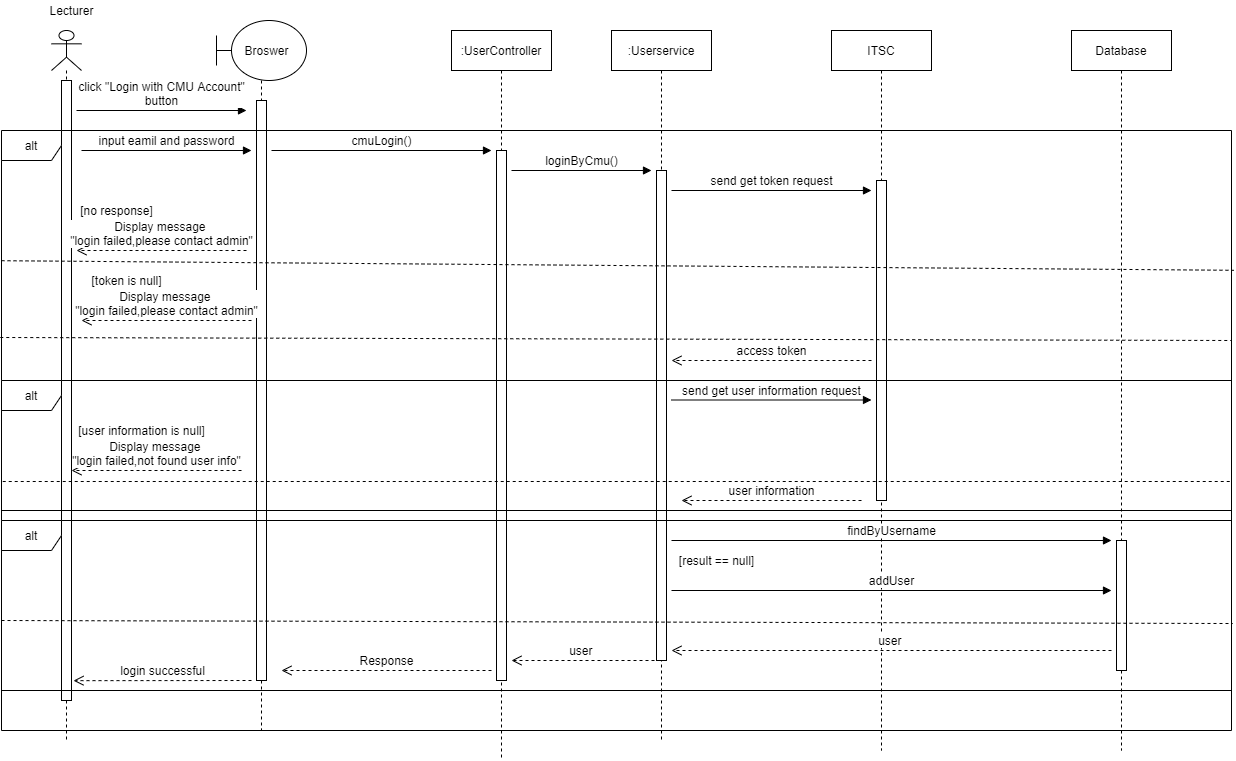
Description: Used to store file record history

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO. | Name | Description | Type | Size | Key |
| 1 | id | This attribute is the id of the file\_record | Int | 11 | PK |
| 2 | pdf\_path | This attribute is the pdf path of the file\_record | varchar | 11 |  |
| 3 | pdf\_img\_dir | This attribute is the img path of the file\_record | varchar | 64 | - |
| 4 | talbe\_img\_dir | This attribute is the table img path of the file\_record | Varchar | 32 | - |
| 5 | excel\_path | This attribute is the excel path of the file\_record | Varchar | 16 | - |
| 6 | created\_time | This attribute is the created time of the file\_record | Datetime | - | - |
| 7 | modified\_time | This attribute is the modified time of the file\_record | Datetime | - | - |
| 8 | created\_user | This attribute is the created user of the file\_record | Int | 11 | FK |
| 9 | modified\_user | This attribute is the modified user of the file\_record | int | 11 | - |

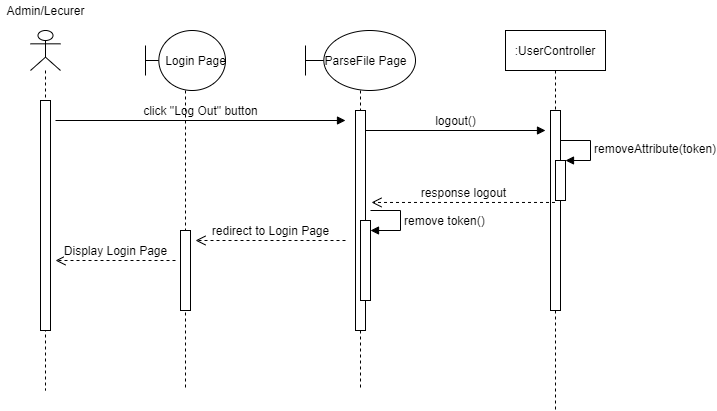
3.3 Sequence Diagram

3.3.1 Authentication system

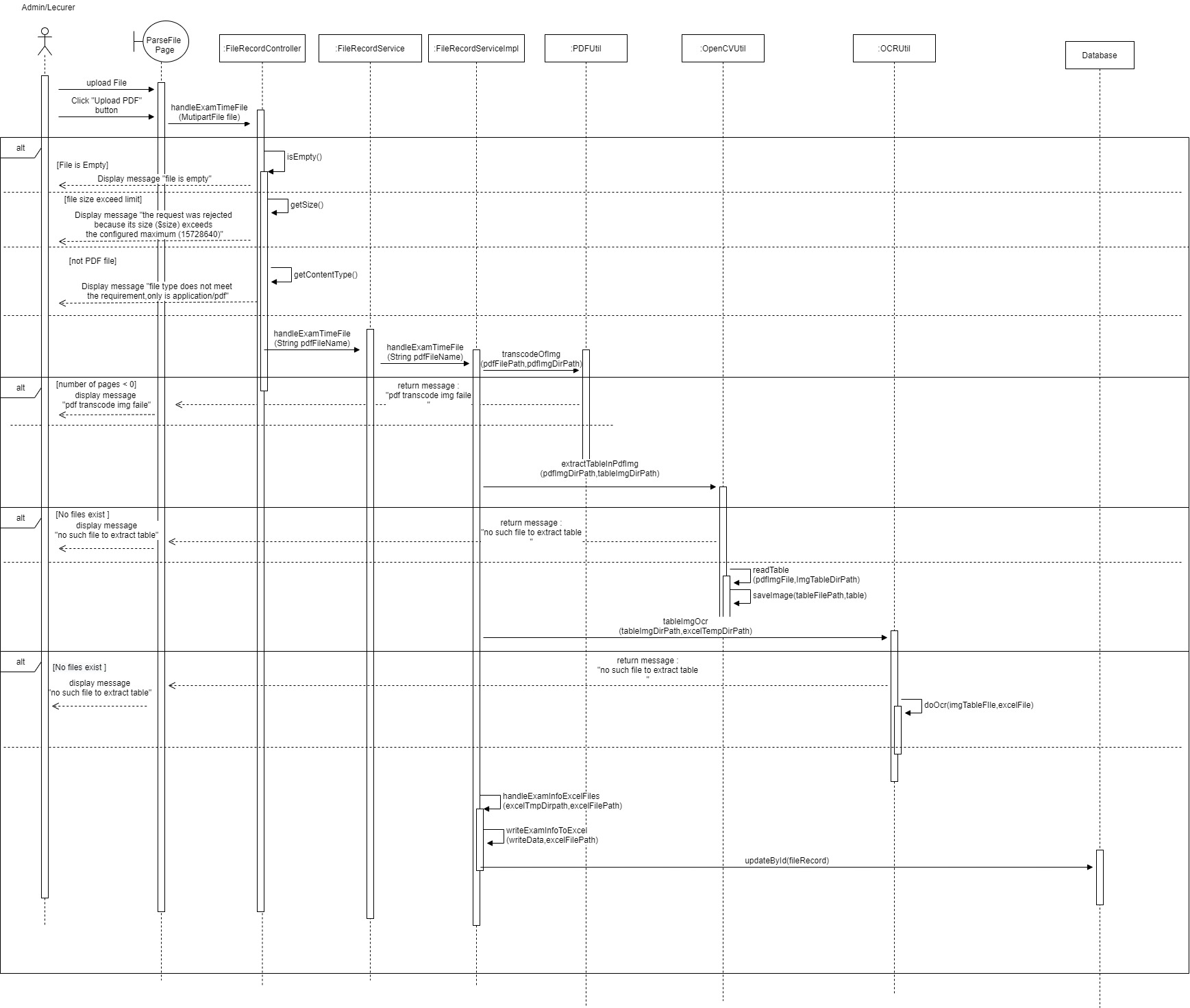
SD-01 Admin Login

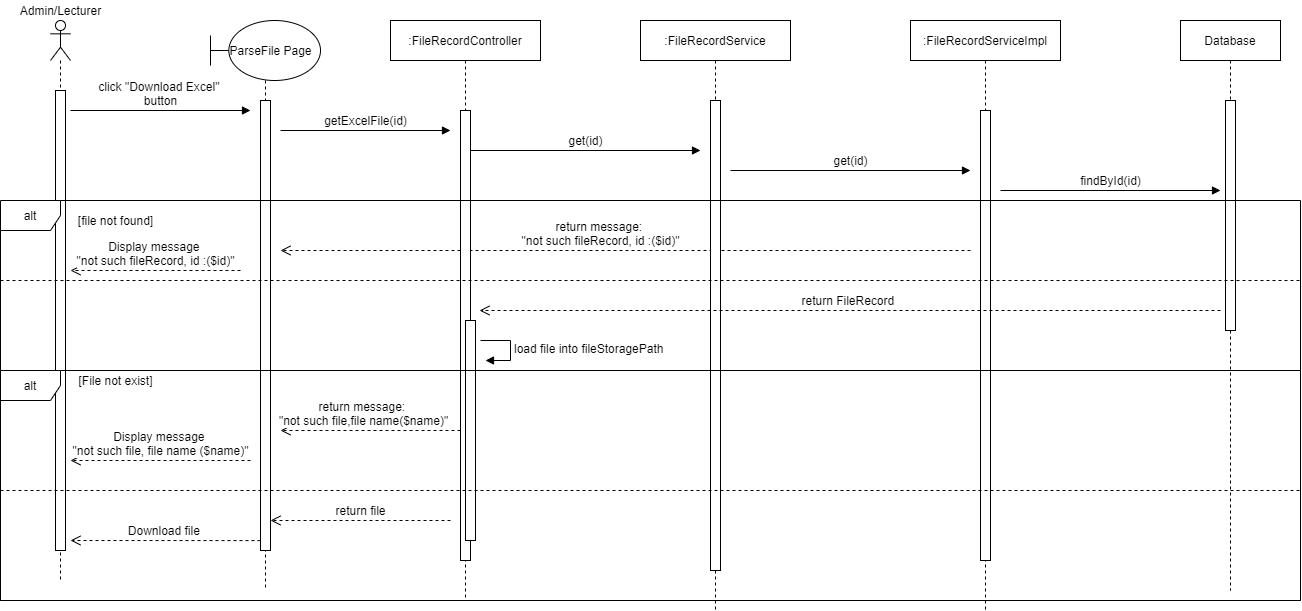
SD-02 Lecturer Login

SD-03 Admin/Lecturer log out

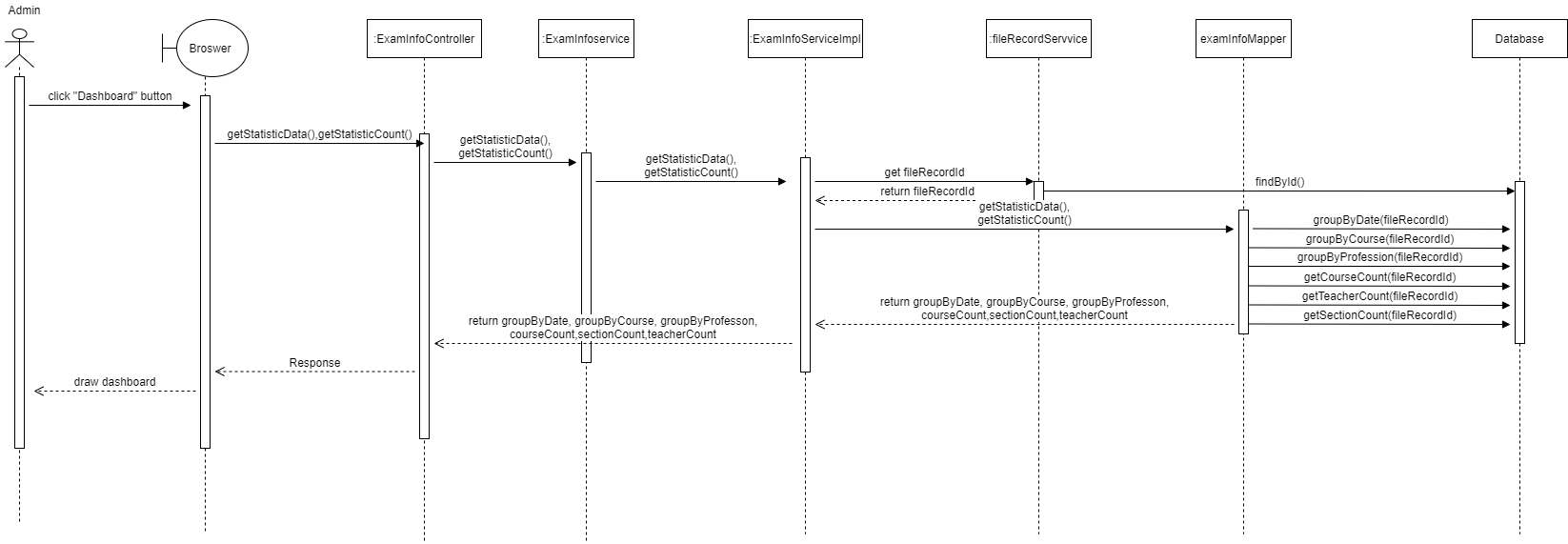


3.3.2 Table and cell detection and text extraction and parsing system

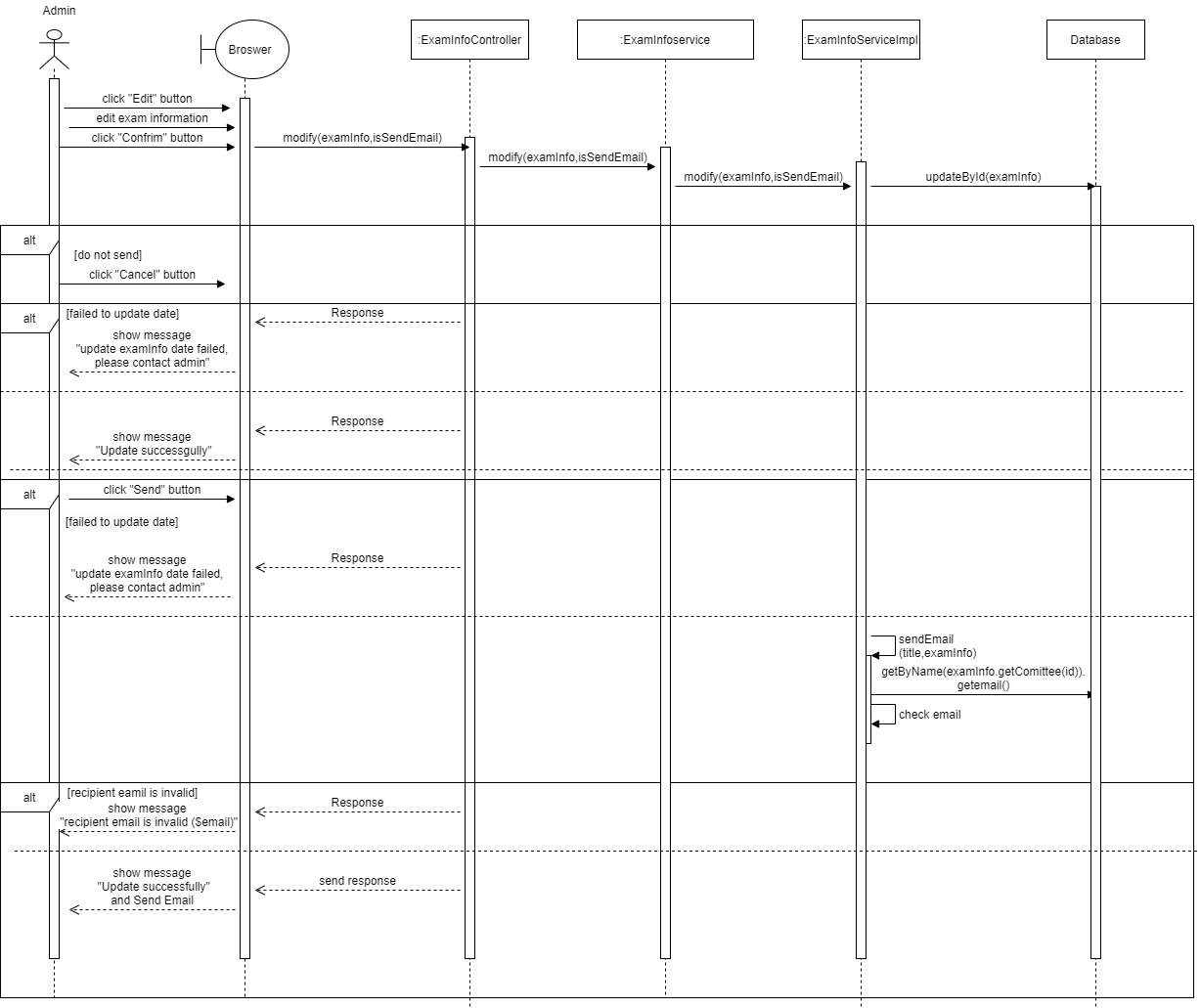
SD-04 Parse CAMT timetable PDF

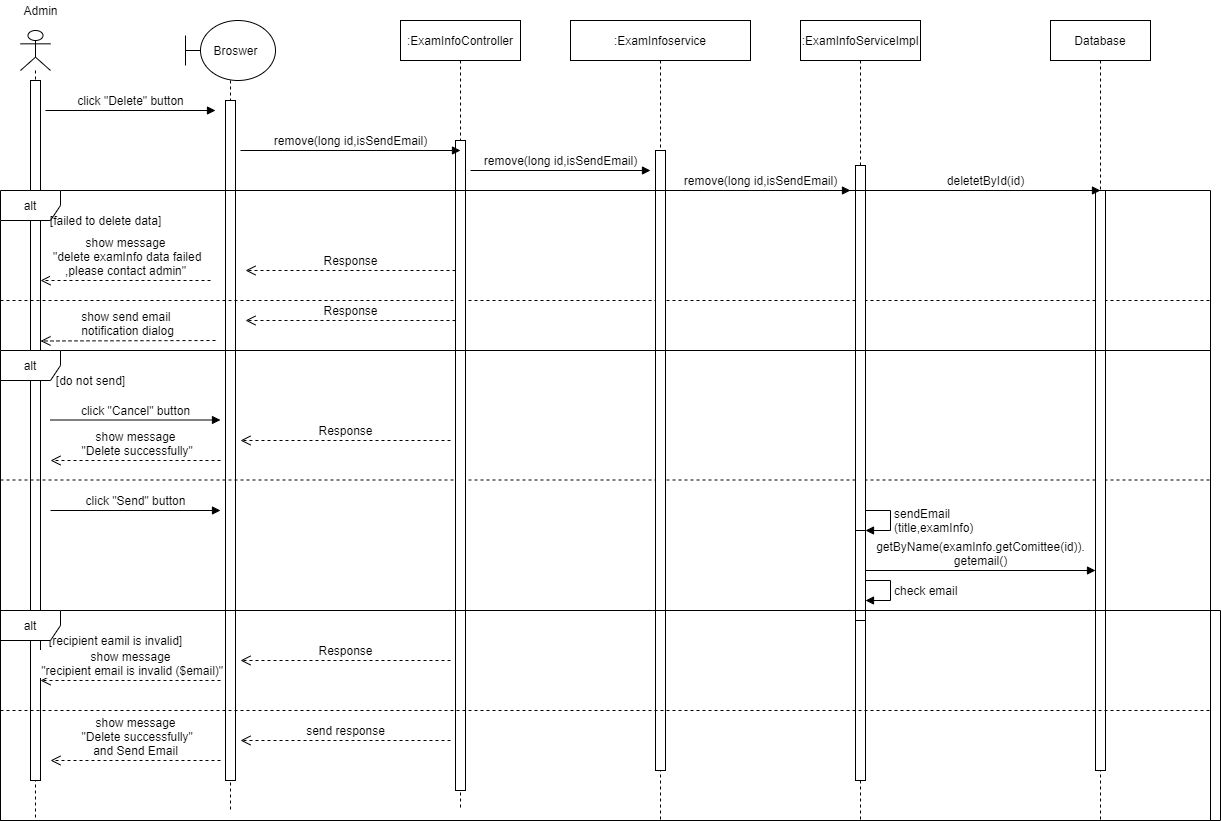
SD-05 Download Excel

3.3.3 Dashboard System

SD-06 Admin/Lecturer view dashboard

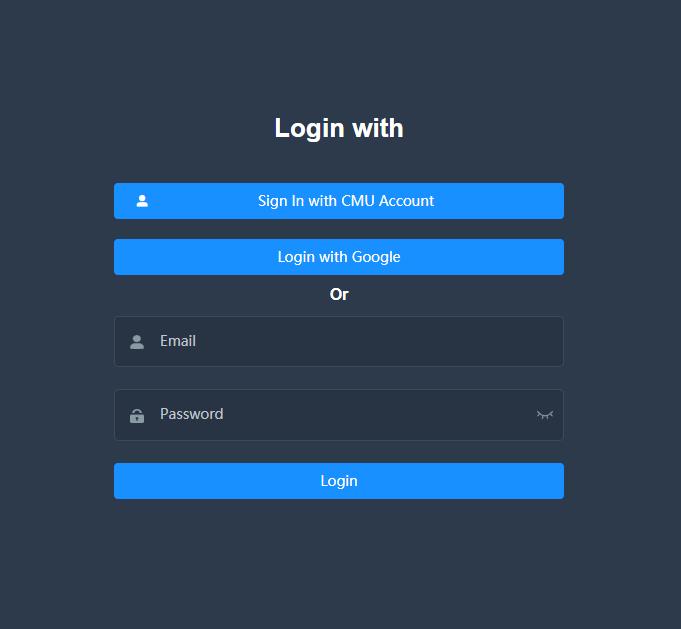
3.4.2 Notification system sequence diagrams

SD-07 Edit exam information

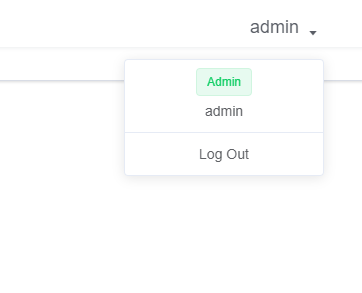
SD-08 Delete exam information

Chapter Four | User Interface Design

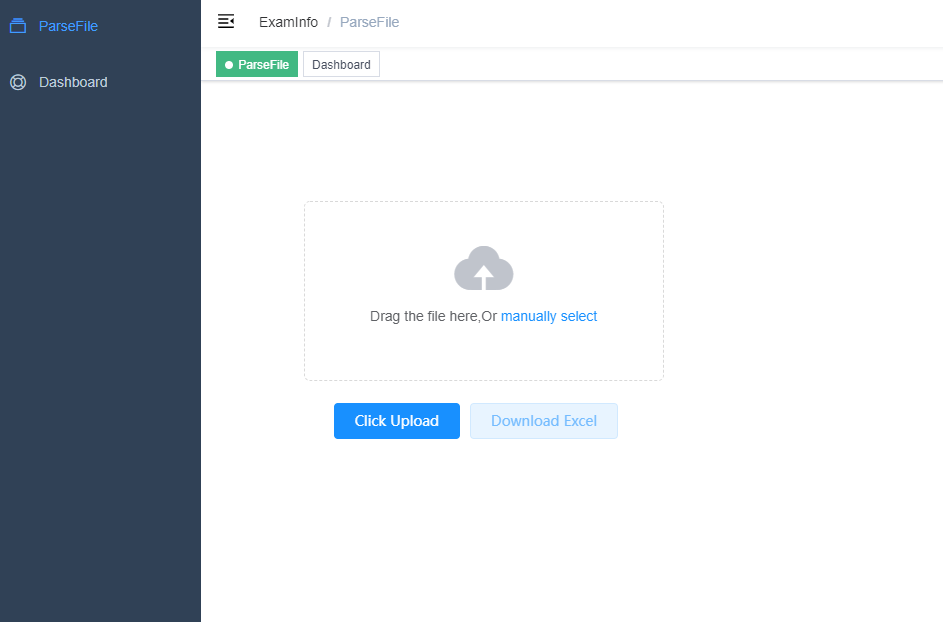
4.1 UI-01 Login Page



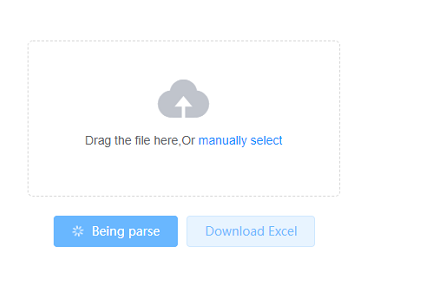
4.2 UI-02 Logout Page



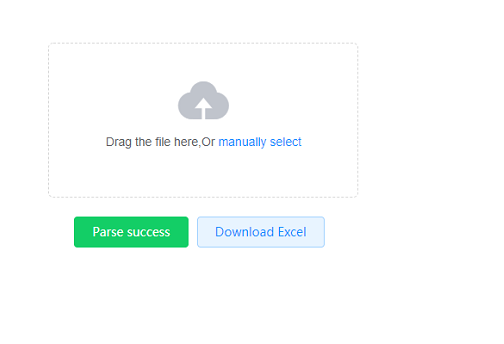
4.3 UI-03 ParseFile Page



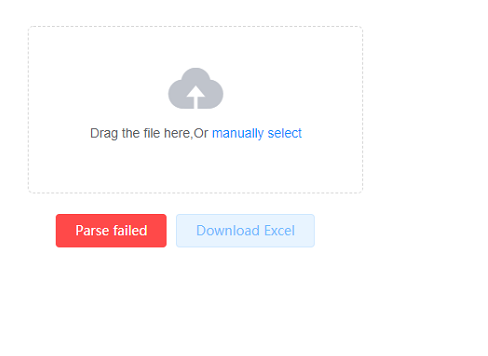
Being parse

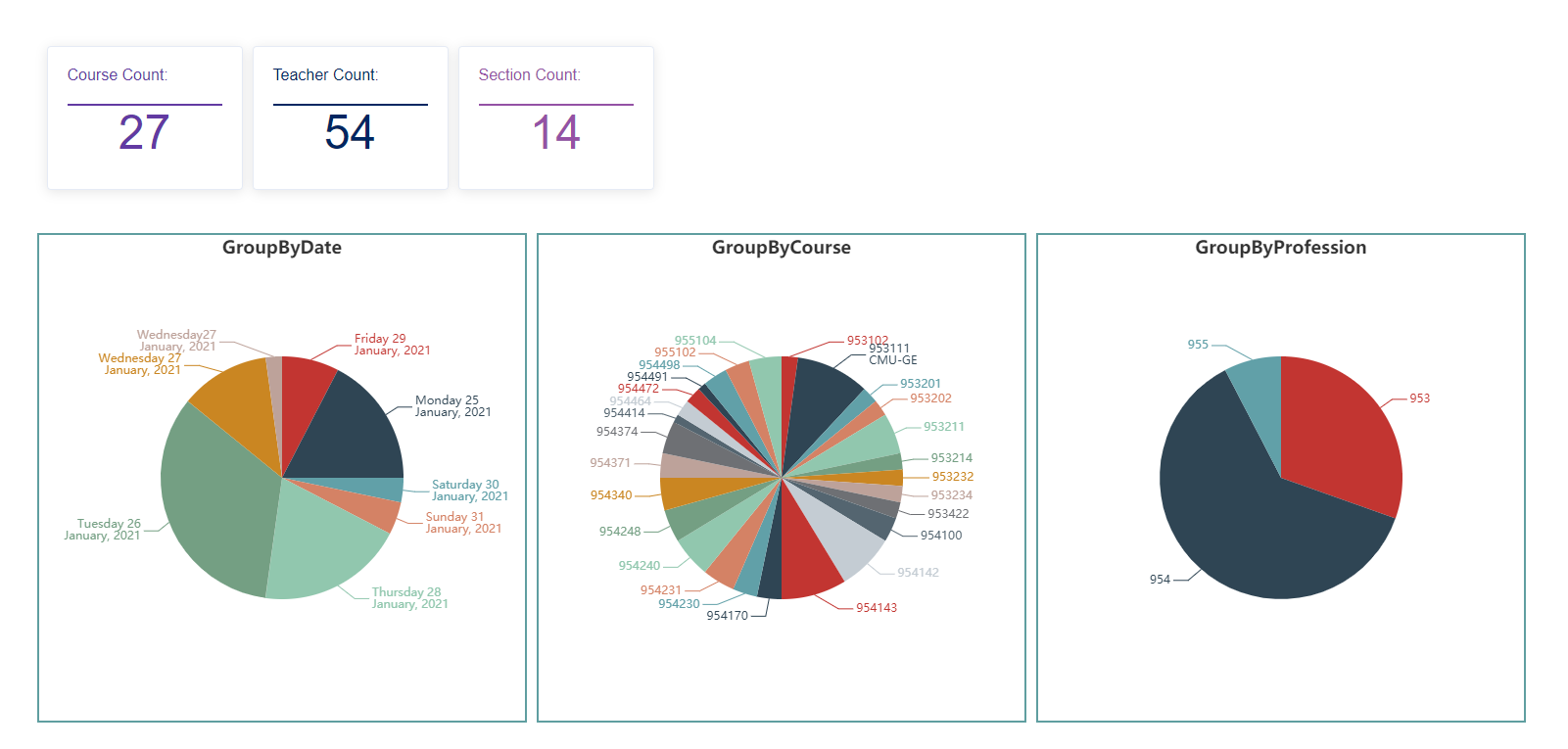


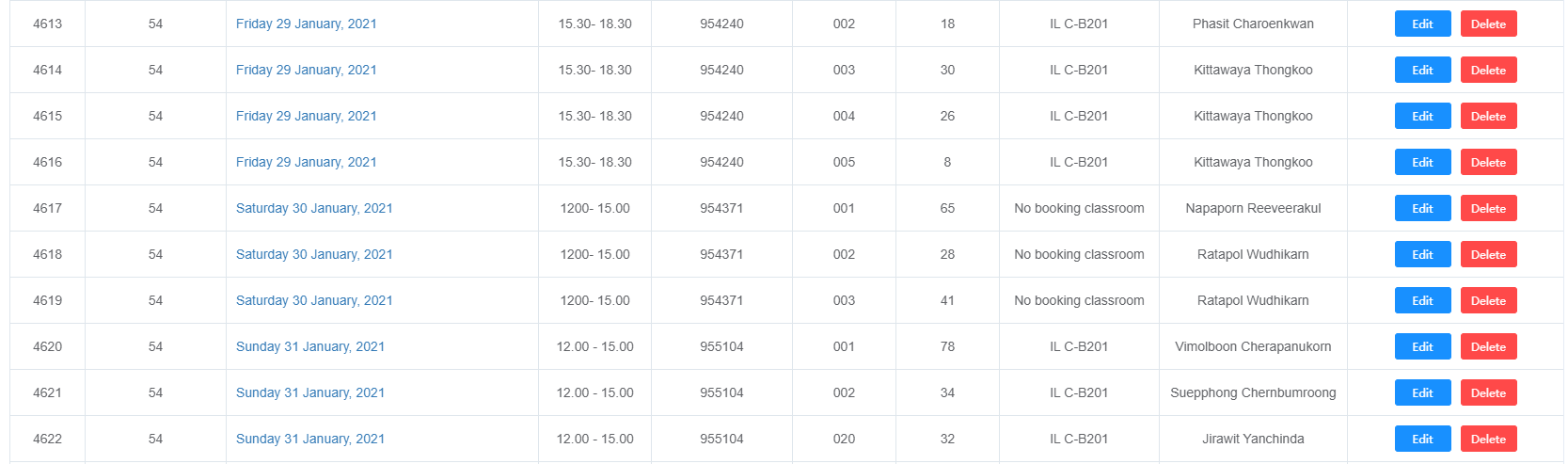
Parse success



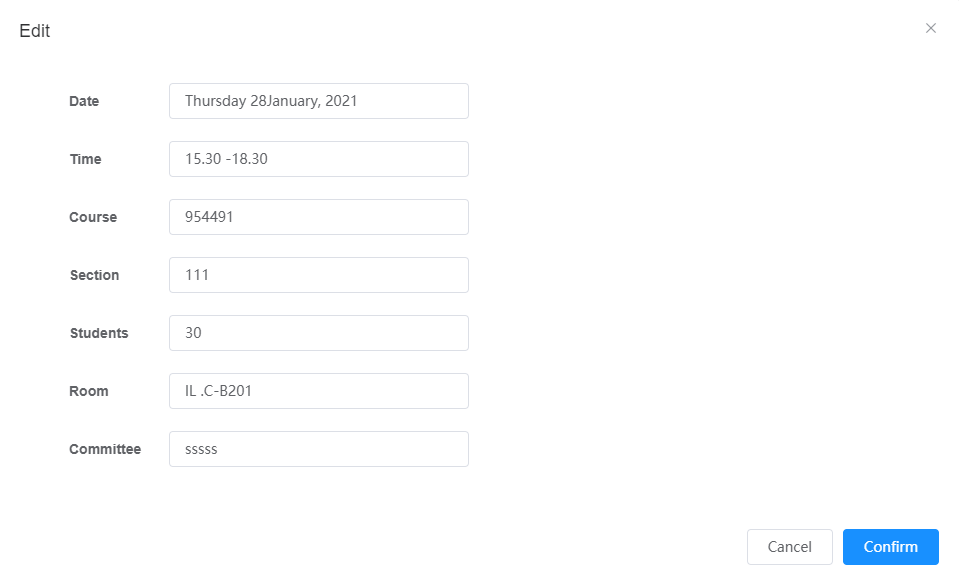
Parse failed



4.4 UI-04 Dashboard Page

4.5 UI-05 ExamInfoList Page

4.6 UI-06 Edit examinfo Page



4.7 UI-07 Send Email page

