Software Requirements Specification

for

CRM Student Admissions

**Version 1.1 approved**

**Prepared by MinhHN**

**SE1736**

**10/10/202**

**Table of Contents**

[1. Introduction 1](#_Toc150250717)

[1.1 Purpose 1](#_Toc150250718)

[1.2 Document Conventions 1](#_Toc150250719)

[1.3 Project Scope 1](#_Toc150250720)

[2. Overall Description 2](#_Toc150250721)

[2.1 Product Perspective 2](#_Toc150250722)

[2.2 User Classes and Characteristics 2](#_Toc150250723)

[2.3 Operating Environment 4](#_Toc150250724)

[4](#_Toc150250725)

[2.4 Design and Implementation Constraints 4](#_Toc150250726)

[i. 3.2.1 Hardware Requirements 4](#_Toc150250727)

[ii. 3.1.2 Software Requirements 4](#_Toc150250728)

[*b.* 4](#_Toc150250729)

[2.5 Assumptions and Dependencies 4](#_Toc150250730)

[3. System Features 6](#_Toc150250731)

[3.1 Login with Google 10](#_Toc150250732)

[3.2 <Student> Send Admission Form 11](#_Toc150250733)

[3.3 <Student> Manage Admission Form 12](#_Toc150250734)

[<Student> Contact Support 19](#_Toc150250735)

[3.9 <Administrator> Backup Data 22](#_Toc150250736)

[3.10 <Administrator> Configure Setting 23](#_Toc150250737)

[<Administrator> Manage Role 25](#_Toc150250738)

[<Administrator> Manage Account 29](#_Toc150250739)

[<Staff> Support Student 33](#_Toc150250740)

[<Staff> Manage Admission Form 36](#_Toc150250741)

[<Staff> Manage Marketing Item 39](#_Toc150250742)

[<Staff> Manage Student 43](#_Toc150250743)

[3.35 <Manager> Generate Report 46](#_Toc150250744)

[<Manager> Manage Major 47](#_Toc150250745)

[<Manager> Manage Curriculum 50](#_Toc150250746)

[<Manager> Manage Staff 54](#_Toc150250747)

[4. Data Requirements 56](#_Toc150250748)

[4.1 Logical Data Model 56](#_Toc150250749)

[4.2 Data Dictionary 57](#_Toc150250750)

[i. Application 57](#_Toc150250751)

[ii. Appointment\_Schedule 57](#_Toc150250752)

[iii. Campaign 58](#_Toc150250753)

[iv. Campaign\_Staff 58](#_Toc150250754)

[v. Communication 58](#_Toc150250755)

[vi. Contract 58](#_Toc150250756)

[vii. Course 59](#_Toc150250757)

[viii. Employee 59](#_Toc150250758)

[ix. Event 59](#_Toc150250759)

[x. Financial 59](#_Toc150250760)

[xi. Major 60](#_Toc150250761)

[xii. Nationality 60](#_Toc150250762)

[xiii. Ranking 60](#_Toc150250763)

[xiv. Report 60](#_Toc150250764)

[xv. Role 60](#_Toc150250765)

[xvi. Student 61](#_Toc150250766)

[xvii. Student\_Nationality 61](#_Toc150250767)

[xviii. System\_Settings 61](#_Toc150250768)

[xix. Task 61](#_Toc150250769)

[xx. Tuition 62](#_Toc150250770)

[4.3 Reports 62](#_Toc150250771)

[4.4 Data Acquisition, Integrity, Retention, and Disposal 62](#_Toc150250772)

[5. External Interface Requirements 62](#_Toc150250773)

[5.1 User Interfaces 62](#_Toc150250774)

[5.2 Software Interfaces 62](#_Toc150250775)

[5.3 Hardware Interfaces 62](#_Toc150250776)

[5.4 Communications Interfaces 63](#_Toc150250777)

[6. Quality Attributes 63](#_Toc150250778)

[6.1 Usability 63](#_Toc150250779)

[*6.2* *Performance* 63](#_Toc150250780)

[6.3 Security 63](#_Toc150250781)

[6.4 Safety 63](#_Toc150250782)

[7. Internationalization and Localization Requirements 63](#_Toc150250783)

[8. Other Requirements 64](#_Toc150250784)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| SWR-Assignment | 10/10/2023 | Created | 1.0 |
| SWR-Assignment | 24/10/2023 | Update Content | 1.05 |
| SWR-Assignment | 27/10/2023 | Updated Content, Added some GUI preview. | 1.1 |
|  |  |  |  |

# Introduction

## Purpose

This Software Requirements Specification (SRS) document describes the requirements for the Student Admissions Support System (SASS), version 1.0. The SASS is a customer relationship management (CRM) system that will be used to manage the admissions process at FPT University.

The intended audience for this document includes:

* Developers
* Project managers
* Testers
* Business Analyst

## Document Conventions

This document uses the following standards and typographical conventions:

* Text styles:
  + **Bold**: Used to highlight important text, such as requirement identifiers and section headings.
  + **Italic**: Used to highlight additional information or to indicate that a term is being defined.
  + **Underlined**: Used to indicate hyperlinks

## Project Scope

The SASS is a CRM system that will be used to manage the admissions process at FPT University. The system will help FPT University to:

Increase student enrollment rates

Improve communication and engagement with prospective students

Streamline the admissions process to increase efficiency and reduce manual efforts

Enhance data accuracy and availability for better decision-making

Provide personalized support to prospective students throughout the admissions journey

The initial release of the SASS will focus on the core features of inquiry management and online application processing. This will include the ability to track and manage inquiries, process online applications, and send automated emails to applicants.

Future releases of the SASS will expand the functionality to include document management, communication tools, and reporting and analytics. These additional features will enhance the overall efficiency and effectiveness of the admissions process.

# Overall Description

The Student Admissions Support System (SASS) is a new CRM system that will be used to manage the admissions process at FPT University. The SASS will replace the existing manual and paper-based admissions process, which is inefficient and time-consuming. The SASS will also integrate with other systems at FPT University, such as the student information system and the website.

## Product Perspective

The Student Admissions Support System (SASS) is a new CRM system that will be used to manage the admissions process at FPT University. The SASS will replace the existing manual and paper-based admissions process, which is inefficient and time-consuming. The SASS will also integrate with other systems at FPT University, such as the student information system and the website.

## User Classes and Characteristics

### 

|  |  |
| --- | --- |
| Primary Actor | Use Case |
| Unauthorized User | 1. Login with Google |
| Student | 1. Send Admission Form   Manage Admission Form   1. Update Admission Form 2. View Admission Form List 3. View Admission Form Details 4. Cancel Admission Form   Contact Support   1. Send Message Request 2. Send Support Ticket |
| Administrator | 1. Backup Data 2. Configure Setting   Manage Role   1. Create Role 2. Update Role 3. View Roles 4. Disable Role   Manage Account   1. Create Account 2. Update Account 3. View Account List 4. View Account Details 5. Disable Account |
| Staff | Support Student   1. Handle Support Ticket 2. Handle Message Request   Manage Admission Form   1. Update Admission Form 2. View Admission Form List 3. View Admission Form Details 4. Cancel Admission Form   Manage Marketing Item   1. Create Item 2. Update Item 3. View Items 4. Disable Item   Manage Student   1. Import/Export Student File 2. Update Student 3. Disable Student 4. View Student List 5. View Student Details |
| Manager | 1. Generate Report   Manage Major   1. Create Major 2. Update Major 3. View Majors 4. Disable Major   Manage Curriculum   1. Import/Export Curriculum File 2. Update Curriculum 3. Disable Curriculum 4. View Curriculum List 5. View Curriculum Details   Manage Staff   1. View Staff List 2. View Staff Details 3. Update Staff |

## Operating Environment

## 

## Design and Implementation Constraints

### 3.2.1 Hardware Requirements

|  |  |  |
| --- | --- | --- |
| PC | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| Operating System | Windows 7 | Windows 10 |
| Computer Processor | Intel® Core i3 1.4GHz | Intel® Core i5 2.50GHz |
| Computer Memory | 1GB RAM | 2GB RAM or more |
| Web Browser | Chromes (v69 or higher) | Chrome latest stable version |

### 3.1.2 Software Requirements

|  |  |  |
| --- | --- | --- |
| Software | Name / Version | Description |
| Operation System | Windows 7, Windows 10 | Operating system and platform for development |
| Web browser | Chrome 69 or above | For web app |
| Android System | Android 0 or higher | For mobile app |

## 

## Assumptions and Dependencies

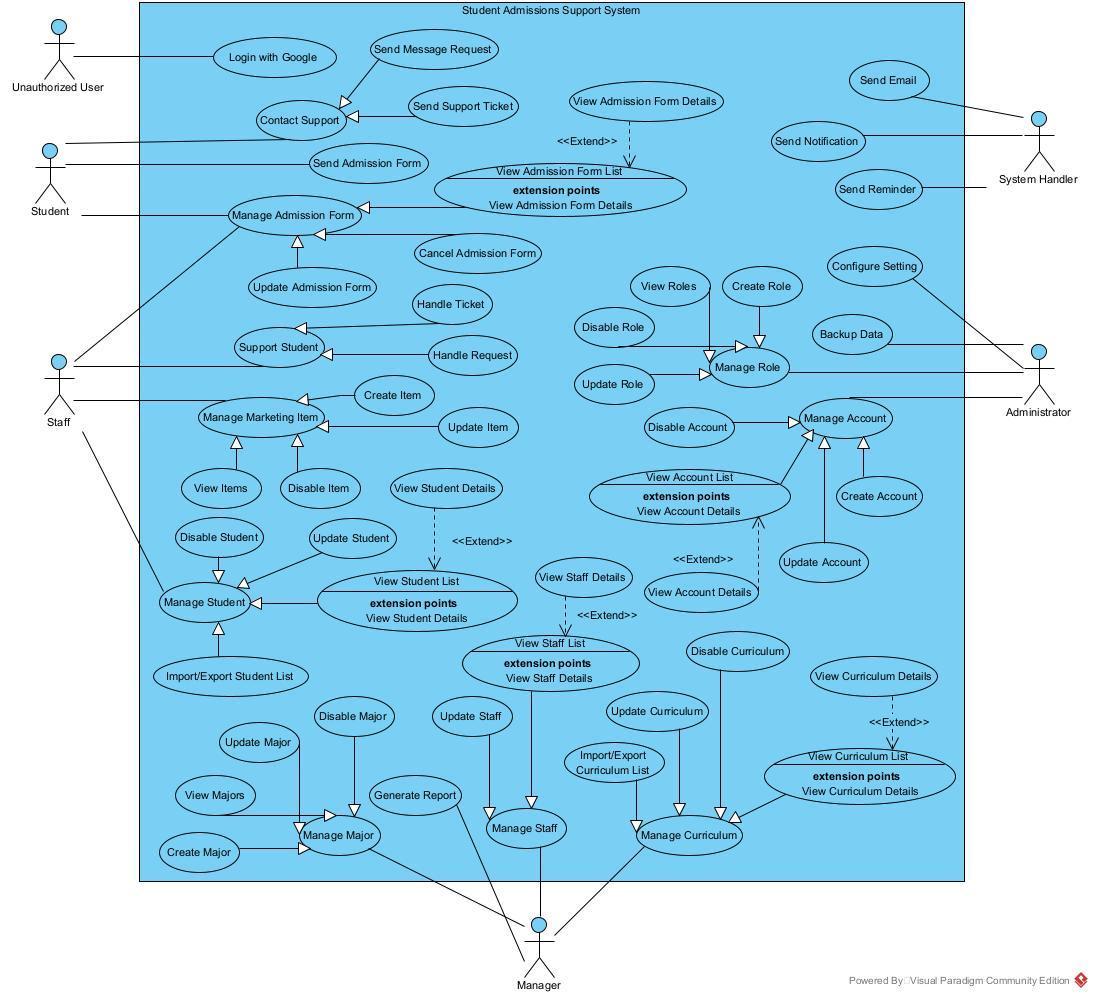
The following assumptions are made about the development and deployment of the SASS:

* The SASS will be integrated with the student information system and the website using existing APIs.
* The SASS will be hosted on FPT University's servers.
* FPT University has the necessary hardware and software resources to support the SASS.

The following dependencies exist for the SASS:

* The student information system and the website must be available and accessible.
* FPT University must have a reliable internet connection.

# System Features

a, Use case diagram:

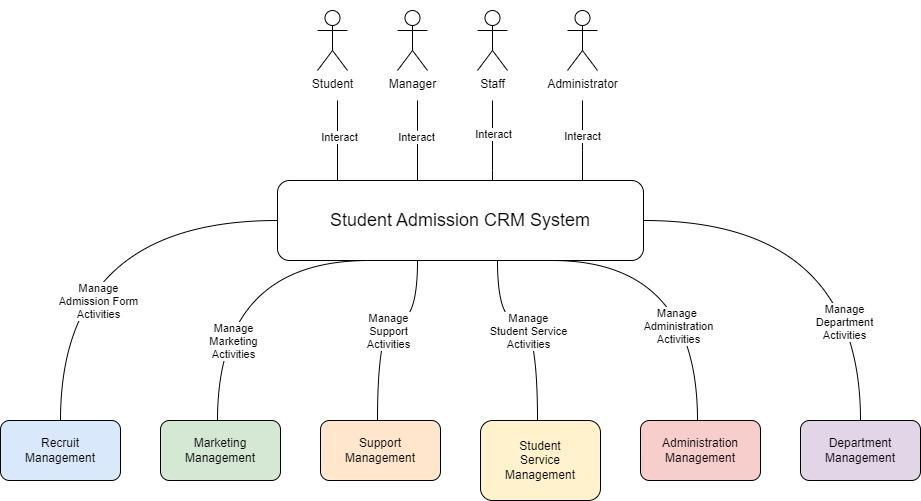
b, Features tree:

A diagram of a company's work flow

Description automatically generated

c, Context diagram

A screenshot of a computer

Description automatically generatedd, Database diagram:

e, Screen flow:

A diagram of a software application

Description automatically generated

### Login with Google

A screenshot of a computer

Description automatically generated a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Login with Google |
| Created By: | MinhHN |
| Date Created: | 10/10/2023 |
| Description: | User can log into the system through a Google mail provided by FPT University |
| Primary Actor: | Unauthorized User |
| Trigger: | The user clicks the "Login with Google” button. |
| Business rule: | Email is authorized in FPT University system, and it must end in “@fpt.edu.vn” or “@fe.edu.vn”. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is not logged into the system |
| Postconditions: | 1. Success: The user login to the system successfully  2. Fail: Cannot use email to log into the system. |
| Main Flow: | 1. Users click the "Login with Google” button.  2. The system shows a pop-up for user to choose the email address they want to log in with.  3. The user chooses an email.  4. The system checks validation and whether it is authorized in the system database.  5. The system redirect user to their authorized home page. |
| Alternative Flows: | N/A |
| Exception: | 4a. The user's chosen email does not match with data in the system.  1. The system displays an error message "This email does not belong to FPT University." |

### <Student> Send Admission Form

A screenshot of a computer

Description automatically generated a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Send Admission Form |
| Created By: | MinhHN |
| Date Created: | 10/10/2023 |
| Description: | Students apply for FPT University admissions |
| Primary Actor: | Student |
| Trigger: | The user clicks the "Send application” button |
| Business rule: | * The system should offer user-friendly interface, application form, and uploading documents. * Students upload required documents in specified formats. * Students must review and verify completed applications before submitting. * System processes application, records timestamp, and sends confirmation to student. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Student”. |
| Postconditions: | 1. Success: Student submits admissions application successfully.  2. Fail: Student make a mistake, or an error occur while submitting admissions application. |
| Main Flow: | 1. Student clicks the " Send application” button.  2. The system offers a user-friendly application form for students to enter personal information, academic background, desired program, and supporting documents.  3. Student completes fields, uploads supporting documents, and ensures accurate information and click “Submit” button.  4. The system validates information, fills required fields, and uses proper formatting. Error messages guide students to correct issues and upload supporting documents in specified formats.  5. System processes application, records timestamp, confirms successful submission to student.  6. System sends student confirmation of successful application submission. |
| Alternative Flows: | N/A |
| Exception: | 4a. The student provides invalid or inappropriate information.  1. System displays error message indicating required information. |

### <Student> Manage Admission Form

Update Admission Form

a, GUI

#### A screenshot of a computer Description automatically generated b, Use case specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Update Admission Form |
| Created By: | MinhHN |
| Date Created: | 10/10/2023 |
| Description: | Students edit admission form information sent to FPT University |
| Primary Actor: | Student |
| Trigger: | The user clicks the "Edit application” button. |
| Business rule: | * The system should offer user-friendly interface, application form, and uploading documents. * Students upload required documents in specified formats. * Students should access admissions process details, including application forms and packets. * Real-time application status updates, including submission, verification, and admission decisions. * System should provide student document submission, assistance, and application inquiries features. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | 1. The user is logged into the system as “Student”. 2. User has sent an admission form to FPT University. 3. Admission Form status is “Pending” or “Review”. |
| Postconditions: | 1. Success: Student can view admissions application successfully.  2. Fail: Student cannot access or view admission form details. |
| Main Flow: | 1. Student clicks the "Edit application” button.  2. System provides student document submission, support, inquiry, and communication options.  3. Student can submit documents, request support, and inquire about application |
| Alternative Flows: | N/A |
| Exception: | 3a. The student provides invalid or inappropriate information.   1. System displays error message indicating required information. |

#### View Admission Form List

A screenshot of a computer

Description automatically generated a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Admission Form List |
| Created By: | MinhHN |
| Date Created: | 12/10/2023 |
| Description: | Student can view the list of admission forms that student has sent. |
| Primary Actor: | Student |
| Trigger: | The user clicks on the "Admission Form" tab on page header. |
| Business rule: | * Ensuring admission form information is securely stored and accessible only to authorized student. * System must validate and handle errors for admission form management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Student”. |
| Postconditions: | 1. Success: Student view admission form successfully.  2. Fail: Student fail in viewing admission form. |
| Main Flow: | 1. Student click on the "Manage Admission Form" tab.  2. Displays a list of existing admission form and management options. |
| Alternative Flows: | N/A |
| Exception: | N/A |

#### View Admission Form Details

A screenshot of a computer

Description automatically generated a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Admission Form Details |
| Created By: | MinhHN |
| Date Created: | 12/10/2023 |
| Description: | Students view admission form sent to FPT University |
| Primary Actor: | Student |
| Trigger: | The user clicks the "View application” button |
| Business rule: | * The system should offer user-friendly interface, application form, and uploading documents. * Students upload required documents in specified formats. * Students should access admissions process details, including application forms and packets. * Real-time application status updates, including submission, verification, and admission decisions. * System should provide student document submission, assistance, and application inquiries features. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Student”.  User has sent an admission form to FPT University. |
| Postconditions: | 1. Success: Student can view admissions application successfully  2. Fail: Student cannot access or view admission form details. |
| Main Flow: | 1. Student clicks the "View application” button.  2. System displays admissions information, including requirements, deadlines, status, updates, and notifications, in a user-friendly interface. |
| Alternative Flows: | 2a. The student monitors application status, verification, and admission decision updates.   1. System offers real-time application status updates, document verification progress, and admission decision notifications.   2b. Student can submit documents, request support, and inquire about application.   1. System provides student document submission, support, inquiry, and communication options. |
| Exception: | N/A |

#### Cancel/Delete Admission Form

a, GUI

A screenshot of a computer

Description automatically generated

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Delete Admission Form |
| Created By: | MinhHN |
| Date Created: | 12/10/2023 |
| Description: | Students cancel admission form sent to FPT University. |
| Trigger: | The user clicks the "Delete” button. |
| Business rule: | * The system should offer a user-friendly application form interface. * Real-time application status updates, including submission, verification, and admission decisions. * System should provide student document submission, assistance, and application inquiries features. |
| Primary Actor: | Student |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | 1. The user is logged into the system as “Student”. 2. User has sent an admission form to FPT University. 3. Admission Form status is not “Approved”. |
| Postconditions: | 1. Success: Student cancel admissions application successfully.  2. Fail: Student cannot cancel admission form. |
| Main Flow: | 1. Student user clicks the "Cancel” button.  2. System displays pending actions for student to complete for application cancellation, ensuring necessary submission steps.  3. Student chooses to proceed to cancel admission.  4. System sends student confirmation of successful application cancellation. |
| Alternative Flows: | N/A |
| Exception: | N/A |

### <Student> Contact Support

#### Send Message Request

A screenshot of a computer

Description automatically generated a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Send Message Request |
| Created By: | MinhHN |
| Date Created: | 12/10/2023 |
| Description: | Students contact FPT University Support Staff |
| Primary Actor: | Student |
| Trigger: | The user clicks the “Chat” button. |
| Business rule: | * Access customer support staff contact information through chat options. * Students deserve timely, satisfactory support from staff to resolve issues. * System should record Support Staff interactions for future reference. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Student”. |
| Postconditions: | 1. Success: Student contact Support Staff successfully.  2. Fail: Student cannot send message successfully |
| Main Flow: | 1. Student clicks the “Chat” button.  2. The system shows a chat pop-up for Student to message with Support Staff.  3. Student submits contact request.  4. System sends contact request to Support Staff.  5. System confirms successful contact request submission. |
| Alternative Flows: | N/A |
| Exception: | 4a. Student contact request delays or unavailable support staff.   1. System informs Student about delay and provides estimated response time. |

#### Send Support Ticket

a, GUI

A screenshot of a computer

Description automatically generated

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Send Support Ticket |
| Created By: | MinhHN |
| Date Created: | 13/10/2023 |
| Description: | Students contact FPT University Support Staff. |
| Primary Actor: | Student |
| Trigger: | The user clicks the “Report Issue” button |
| Business rule: | * Access customer support staff contact information through ticket options. * Students deserve timely, satisfactory support from staff to resolve issues. * System should record Support Staff interactions for future reference. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Student”. |
| Postconditions: | 1. Success: Student contact Support Staff successfully.  2. Fail: Student cannot send ticket successfully. |
| Main Flow: | 1. Student clicks the “Report Issue” button.  2. The system offers a user-friendly application form for students to enter account information, and issue description.  3. Student provides details, account information, and issue description.  4. Student submits contact request  5. System sends ticket request to Support Staff.  6. System confirms successful ticket request submission.  7. System checks if Student requests are addressed by Support Staff.  8. System sends Student notification or displays response. |
| Alternative Flows: | N/A |
| Exception: | 4a. Student ticket request delays or unavailable support staff.   1. System informs Student about delay and provides estimated response time. |

### <Administrator> Backup Data

A screenshot of a computer

Description automatically generateda, GUI

b, Use case specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Backup Data |
| Created By: | MnhHN |
| Date Created: | 13/10/2023 |
| Description: | Administrator can backup data manually or set auto-backup schedule. |
| Primary Actor: | Administrator |
| Trigger: | The user clicks the " Backup Data” tab. |
| Business rule: | * Choose a higher-specified backup schedule if an exception occurs. * Select large or large backup schedules for efficient database instances in production environments. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Admin”. |
| Postconditions: | 1. Success: Administrator backup data successfully.  2. Fail: Administrator cannot backup data or change backup schedule. |
| Main Flow: | 1. Admin clicks the "Backup Data” tab  2. System displays details of backup status and time, including backup management, in a user-friendly interface. |
| Alternative Flows: | 2a. Administrator clicks “Backup now” button.   1. System backup all data corresponding to real-time application and send a notification of backup successful.   2b. System shows a pop-up for user to schedule an automatic time for backing up data.   1. Administrator choose an appropriate time corresponding with system validation. 2. System sends a notification and schedules the time for backup schedule. |
| Exception: | N/A |

### <Administrator> Configure Setting

a, GUI

A screenshot of a computer

Description automatically generated b, Use case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Configure Setting |
| Created By: | MinhHN |
| Date Created: | 13/10/2023 |
| Description: | This use case enables the Administrator to configure the web application's system settings. |
| Primary Actor: | Administrator |
| Trigger: | Administrator clicks the "Configure System Setting" tab on the left side bar. |
| Business rule: | * Administrator should configure system settings for web application customization. * Validate and enforce system settings for application stability and security. * System should handle incorrect configuration values with appropriate feedback. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | User is logged into the system as a “Administrator”. |
| Postconditions: | 1. Success: The requested system settings have been configured successfully.  2. Fail: Cannot configure the setting. |
| Main Flow: | 1. Administrator selects the "Configure System Setting" option.  2. System displays available settings categories.  3. Administrator chooses a category from list.  4. System displays configuration options and values for selected category.  5. Administrator adjusts configuration values for desired settings.  6. System checks and applies changes to settings.  7. System confirms configuration update with success message. |
| Alternative Flows: | N/A |
| Exception: | 2a. System settings categories unavailable.   1. System message shows no available settings.   3a Exception   1. Administrator chooses unavailability or unavailable category. 2. System displays invalid category selection error message.   5a. Administrator provides incorrect configuration values.   1. System displays error message indicating required configuration values. |

### <Administrator> Manage Role

#### Create Role

a, GUI

A screenshot of a computer

Description automatically generated

b, Use case specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Create Role |
| Created By: | MinhHN |
| Date Created: | 14/10/2023 |
| Description: | Administrator can create new roles. |
| Primary Actor: | Administrator |
| Trigger: | The user clicks the “Create Role” button. |
| Business rule: | * Ensuring role information is securely stored and accessible only to authorized administrators. * System must validate and handle errors for role management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Admin”. |
| Postconditions: | 1. Success: Administrator create role successfully.  2. Fail: Administrator fail in creating role. |
| Main Flow: | 1. Clicks on the "Create Role" button.  2. Displays a list of existing Roles  3. Selects the option to add a new role  4. Show form to enter new role details  5. Enter the details of the new role, including name and description.  6. System shows message that the roles creating successfully. |
| Alternative Flows: | N/A |
| Exception: | 1a. User input invalid fields   1. System displays error message indicating required re-entering invalid fields.   2a. Role existed.   1. System displays error message indicating entering another role |

#### Update Role

A screenshot of a computer

Description automatically generateda, GUI

b, Use case specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Update Role |
| Created By: | MinhHN |
| Date Created: | 13/10/2023 |
| Description: | Administrator can update new roles. |
| Primary Actor: | Administrator |
| Trigger: | The user clicks the “Update” icon on a row. |
| Business rule: | * Ensuring role information is securely stored and accessible only to authorized administrators. * System must validate and handle errors for role management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Admin”. |
| Postconditions: | 1. Success: Administrator update role successfully.  2. Fail: Administrator fail in updating role. |
| Main Flow: | 1. Administrator clicks the “Update” icon  2. Show form to edit role information  3. Enter the new details of the role  4. System shows message that the role is edited successfully. |
| Alternative Flows: | N/A |
| Exception: | 1a. User input invalid fields   1. System displays error message indicating required re-entering invalid fields.   2a. Role existed   1. System displays error message indicating entering another role |

#### View Roles

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Roles |
| Created By: | MinhHN |
| Date Created: | 14/10/2023 |
| Description: | Administrator can view list of existing roles. |
| Primary Actor: | Administrator |
| Trigger: | The user clicks on the "Manage Role" tab |
| Business rule: | * Ensuring role information is securely stored and accessible only to authorized administrators. * System must validate and handle errors for role management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Admin”. |
| Postconditions: | 1. Success: Administrator view role successfully.  2. Fail: Administrator fail in viewing role. |
| Main Flow: | 1. Administrator clicks on the "Manage Role" tab.  2. Displays a list of existing role and management options. |
| Alternative Flows: | N/A |
| Exception: | N/A |

#### Disable Role

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Disable Role |
| Created By: | MinhHN |
| Date Created: | 14/10/2023 |
| Description: | Administrator can disable role. |
| Primary Actor: | Administrator |
| Trigger: | The user clicks the “Disable” icon on a row. |
| Business rule: | * Ensuring role information is securely stored and accessible only to authorized administrators. * System must validate and handle errors for role management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Admin”. |
| Postconditions: | 1. Success: Administrator disable role successfully  2. Fail: Administrator fail in disabling role |
| Main Flow: | 1. Administrator clicks the “Update” icon  2. System shows message that the role is deactivated, and services are also deactivated. |
| Alternative Flows: | N/A |
| Exception: | N/A |

### <Administrator> Manage Account

#### Create Account

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Create Account |
| Created By: | MinhHN |
| Date Created: | 15/10/2023 |
| Description: | Administrator can create a new account in the system. |
| Primary Actor: | Administrator |
| Trigger: | The user clicks the “Create account” button |
| Business rule: | * Ensuring Account information is securely stored and accessible only to authorized administrators. * System must validate and handle errors for account management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Admin”. |
| Postconditions: | 1. Success: Administrator create an account successfully  2. Fail: Administrator fail in creating Account. |
| Main Flow: | 1. Administrator clicks the “Create account” button.  2. System show form to input account information  3. User enter the new details of the account  4. System shows message that the account is created successfully. |
| Alternative Flows: | N/A |
| Exception: | 3a. User input invalid fields   1. System displays error message indicating required re-entering invalid fields.   4a. Account existed.   1. System displays error message indicating entering another Account. |

#### Update Account

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Update Account |
| Created By: | MinhHN |
| Date Created: | 15/10/2023 |
| Description: | Administrator can update existing account. |
| Primary Actor: | Administrator |
| Trigger: | The user clicks the “Update” icon on a row |
| Business rule: | * Ensuring Account information is securely stored and accessible only to authorized administrators. * System must validate and handle errors for account management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Admin”. |
| Postconditions: | 1. Success: Administrator update account successfully  2. Fail: Administrator fail in updating Account. |
| Main Flow: | 1. Administrator clicks the “Update” icon  2. System show form to edit account information  3. User enter the new details of the account  4. System shows message that the account is edited successfully. |
| Alternative Flows: | N/A |
| Exception: | 3a. User input invalid fields   1. System displays error message indicating required re-entering invalid fields.   4a. Account existed.   1. System displays error message indicating entering another Account. |

#### View Account List

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Account List |
| Created By: | MinhHN |
| Date Created: | 15/10/2023 |
| Description: | Administrator can view list of existing accounts. |
| Primary Actor: | Administrator |
| Trigger: | The user clicks on the "Manage Account" tab. |
| Business rule: | * Ensuring Account information is securely stored and accessible only to authorized administrators. * System must validate and handle errors for account management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Admin”. |
| Postconditions: | 1. Success: Administrator view account successfully  2. Fail: Administrator fail in viewing Account. |
| Main Flow: | 1. Administrator clicks on the "Manage Account" tab.  2. Displays a list of existing account and management options. |
| Alternative Flows: | N/A |
| Exception: | N/A |

#### View Account Details

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Account Details |
| Created By: | MinhHN |
| Date Created: | 17/10/2023 |
| Description: | Administrator can view details of existing Account. |
| Primary Actor: | Administrator |
| Trigger: | The user clicks on the "View Details" icon on a row. |
| Business rule: | * Ensuring Account information is securely stored and accessible only to authorized administrators. * System must validate and handle errors for account management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Admin”. |
| Postconditions: | 1. Success: Administrator view account details successfully.  2. Fail: Administrator fail in viewing account details. |
| Main Flow: | 1. Administrator clicks on the "View Details" icon on a row.  2. Displays Account Details page with all information of an account. |
| Alternative Flows: | N/A |
| Exception: | N/A |

#### Disable Account

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Disable Account |
| Created By: | MinhHN |
| Date Created: | 17/10/2023 |
| Description: | Administrator can disable account. |
| Primary Actor: | Administrator |
| Trigger: | The user clicks the “Disable” icon on a row. |
| Business rule: | * Ensuring Account information is securely stored and accessible only to authorized administrators. * System must validate and handle errors for account management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Admin”. |
| Postconditions: | 1. Success: Administrator disable account successfully.  2. Fail: Administrator fail in disabling account. |
| Main Flow: | 1. Administrator clicks the “Disable” icon on a row.  2. System shows message that the account is deactivated, and services are also deactivated. |
| Alternative Flows: | N/A |
| Exception: | N/A |

### <Staff> Support Student

#### Handle Support Ticket

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Handle Support Ticket |
| Created By: | MinhHN |
| Date Created: | 17/10/2023 |
| Description: | Staff manages user support tickets, ensuring efficient customer service. |
| Primary Actor: | Staff |
| Trigger: | Staff clicks “Manage Ticket” on left side bar |
| Business rule: | * Promptly handle support tickets for user assistance. * The system should offer Staff access to essential tools. * Staff should understand user issues, provide accurate solutions, and ensure satisfaction. * Logging and tracking support tickets ensure timely resolution and effective communication. * Staff prioritize support tickets based on urgency and customer impact. * Staff and customers receive notifications throughout support ticket lifecycle. * Escalation procedures essential for handling complex support tickets |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “Support Staff” by Manager. |
| Postconditions: | 1. Success: The support ticket has been resolved, and the user is pleased.  2. Fail: Cannot handle ticket or user is not pleased. |
| Main Flow: | 1. Staff clicks “Manage Ticket” on left side bar.  2. System displays support ticket details, user information, description, attachments.  3. Staff reviews ticket and gathers additional information if needed.  4. System offers communication tools for Staff.  5. Staff gathers user details, aids, and troubleshoots issues  6. System records communication history and updates support ticket status.  7. Staff analyze issues and offer solutions  8. Staff can document solution or workaround.  9. Staff communicate solutions to users, ensuring understanding.  10. System records communication history and updates support ticket status.  11. Staff confirm issue resolution and address user questions.  12. System records support ticket resolution status.  13. Staff closes support ticket.  14. System closes support ticket status and notifies user. |
| Alternative Flows: | N/A |
| Exception: | 2a Incomplete or missing support ticket details.   1. System prompts Staff for user information.   4a. System communication tools and options not functioning properly.   1. System displays error message, suggests alternative communication methods.   5a. Staff struggles to reach user due to unresponsiveness.   1. Staff can ticket user responses via message or email   7a. Staff struggles to find user issue solution.   1. System suggests escalating support ticket to higher level or team involvement.   9a. User dissatisfied with solution or workaround provided.   1. System suggests Staff seek assistance from experienced supervisor.   11a. User confirms issue unresolved or has additional questions or concerns.   1. System prompts Staff to address user needs. |

#### Handle Message Request

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Handle Support Request |
| Created By: | MinhHN |
| Date Created: | 17/10/2023 |
| Description: | Staff manages user support requests, ensuring efficient customer service. |
| Primary Actor: | Staff |
| Trigger: | Staff clicks “Manage Request” on left side bar. |
| Business rule: | * Promptly handle support requests for user assistance. * The system should offer Staff access to essential tools. * Staff should understand user issues, provide accurate solutions, and ensure satisfaction. * Logging and tracking support requests ensure timely resolution and effective communication. * Staff prioritize support requests based on urgency and customer impact. * Staff and customers receive notifications throughout support request lifecycle. * Escalation procedures essential for handling complex support requests. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “Student Service Staff” by Manager. |
| Postconditions: | 1. Success: The support request has been resolved, and the user is pleased  2. Fail: Cannot handle request or user is not pleased. |
| Main Flow: | 1. Staff clicks “Manage Request” on left side bar.  2. System displays support request details, user information, description, attachments.  3. Staff reviews request and gathers additional information if needed.  4. System offers communication tools for Staff.  5. Staff gathers user details, aids, and troubleshoots issues  6. System records communication history and updates support request status.  7. Staff analyze issues and offer solutions.  8. Staff can document solution or workaround.  9. Staff communicate solutions to users, ensuring understanding.  10. System records communication history and updates support request status.  11. Staff confirm issue resolution and address user questions.  12. System records support ticket resolution status.  13. Staff closes support request.  14. System closes support request status and notifies user. |
| Alternative Flows: | N/A |
| Exception: | 2a. Incomplete or missing support request details.   1. System prompts Staff for user information.   4a. System communication tools and options not functioning properly.   1. System displays error message, suggests alternative communication methods.   5a. Staff struggles to reach user due to unresponsiveness.   1. Staff can request user responses via message or email.   7a. Staff struggles to find user issue solution.   1. System suggests escalating support request to higher level or team involvement.   9a. User dissatisfied with solution or workaround provided.   1. System suggests Staff seek assistance from experienced supervisor.   11a. User confirms issue unresolved or has additional questions or concerns.   1. System prompts Staff to address user needs. |

### <Staff> Manage Admission Form

#### Update Admission Form

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Approve Admission Form |
| Created By: | MinhHN |
| Date Created: | 18/10/2023 |
| Description: | Staff can approve admission form sent by Student. |
| Primary Actor: | Staff |
| Trigger: | The user changes the form status. |
| Business rule: | * Ensuring admission form information is securely stored and accessible only to authorized staff. * System must validate and handle errors for admission form management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “Recruit Staff” by Manager.  Admission Form status is not “Approved”. |
| Postconditions: | 1. Success: Staff approves admission form details successfully  2. Fail: Staff fail in approving admission form details |
| Main Flow: | 1. Staff changes the form status to “Approved”.  2. The system prompts the user to confirm the status.  3. Staff confirms the status changed.  4. The system change form status.  5. The system displays a success message confirming the form status changed. |
| Alternative Flows: | 1a. Staff changes the form status to “Disapproved”.   1. The system continues with step 2 in the Main Scenario.   1b. Staff changes the form status to “Review”.   1. The system continues with step 2 in the Main Scenario.   2a. Status changed fail   1. The system shows a message about status changed failed. |

#### View Admission Form List

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Admission Form List |
| Created By: | MinhHN |
| Date Created: | 18/10/2023 |
| Description: | Staff can view the list of existing admission forms. |
| Primary Actor: | Staff |
| Trigger: | The user clicks on the "Manage Admission Form" tab. |
| Business rule: | * Ensuring admission form information is securely stored and accessible only to authorized staff. * System must validate and handle errors for admission form management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “Recruit Staff” by Manager. |
| Postconditions: | 1. Success: Staff view admission form successfully.  2. Fail: Staff fail in viewing admission form. |
| Main Flow: | 1. Staff clicks on the "Manage Admission Form" tab.  2. Displays a list of existing admission form and management options. |
| Alternative Flows: | N/A |
| Exception: | N/A |

#### View Admission Form Details

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Admission Form Details |
| Created By: | MinhHN |
| Date Created: | 18/10/2023 |
| Description: | Staff can view details of existing Admission Form |
| Primary Actor: | Staff |
| Trigger: | The user clicks on the "View Details" icon on a row. |
| Business rule: | * Ensuring admission form information is securely stored and accessible only to authorized staff. * System must validate and handle errors for admission form management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “Recruit Staff” by Manager. |
| Postconditions: | 1. Success: Staff view admission form details successfully.  2. Fail: Staff fail in viewing admission form details. |
| Main Flow: | 1. Staff clicks on the "View Details" icon on a row.  2. Displays Admission Form Details page with all information of an admission form. |
| Alternative Flows: | N/A |
| Exception: | N/A |

#### Cancel Admission Form

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Cancel Admission Form |
| Created By: | MinhHN |
| Date Created: | 19/10/2023 |
| Description: | Staff cancel admission form sent to FPT University. |
| Trigger: | The user clicks the "Cancel” button. |
| Business rule: | * The system should offer a user-friendly application form interface. * Real-time application status updates, including submission, verification, and admission decisions. * System should provide staff document submission, assistance, and application inquiries features. |
| Primary Actor: | Staff |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | 1. The user is logged into the system as “Staff”. 2. User has sent an admission form to FPT University. 3. Admission Form status is not “Approved”. |
| Postconditions: | 1. Success: Staff cancel admissions application successfully.  2. Fail: Staff cannot cancel admission form. |
| Main Flow: | 1. Staff user clicks the "Cancel” button.  2. System displays pending actions for user to complete for application cancellation, ensuring necessary submission steps.  3. Staff chooses to proceed to cancel admission.  4. System sends a confirmation of successful application cancellation. |
| Alternative Flows: | N/A |
| Exception: | N/A |

### <Staff> Manage Marketing Item

#### Create Marketing Item

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Create Marketing Item |
| Created By: | MinhHN |
| Date Created: | 20/10/2023 |
| Description: | Staff can create new marketing items. |
| Primary Actor: | Staff |
| Trigger: | The user clicks the “Create Marketing Item” button. |
| Business rule: | * Ensuring marketing item information is securely stored and accessible only to authorized staff. * System must validate and handle errors for marketing item management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “PR Staff” by Manager. |
| Postconditions: | 1. Success: Staff create marketing item successfully.  2. Fail: Staff fail in creating marketing item. |
| Main Flow: | 1. Clicks on the "Create Marketing Item" button  2. Displays a list of existing Marketing Items  3. Selects the option to add a new marketing item  4. Show form to enter new marketing item details  5. Enter the details of the new marketing item, including name and description.  6. System shows message that the marketing items creating successfully. |
| Alternative Flows: | N/A |
| Exception: | 1a. User input invalid fields  1. System displays error message indicating required re-entering invalid fields.  2a. Marketing Item existed  1. System displays error message indicating entering another marketing item. |

#### Update Marketing Item

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Update Marketing Item |
| Created By: | MinhHN |
| Date Created: | 20/10/2023 |
| Description: | Staff can update new marketing items. |
| Primary Actor: | Staff |
| Trigger: | The user clicks the “Update” icon on a row. |
| Business rule: | * Ensuring marketing item information is securely stored and accessible only to authorized staff. * System must validate and handle errors for marketing item management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “PR Staff” by Manager. |
| Postconditions: | 1. Success: Sta Staff update marketing item successfully.  2. Fail: Staff fail in updating marketing item. |
| Main Flow: | 1. Staff clicks the “Update” icon  2. Show form to edit marketing item information  3. Enter the new details of the marketing item  4. System shows message that the marketing item is edited successfully |
| Alternative Flows: | N/A |
| Exception: | 1a. User input invalid fields   1. System displays error message indicating required re-entering invalid fields.   2a. Marketing Item existed   1. System displays error message indicating entering another marketing item. |

#### View Marketing Items

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Marketing Items |
| Created By: | MinhHN |
| Date Created: | 20/10/2023 |
| Description: | Staff can view list of existing marketing items. |
| Primary Actor: | Staff |
| Trigger: | The user clicks on the "Manage Marketing Item" tab. |
| Business rule: | * Ensuring marketing item information is securely stored and accessible only to authorized staff. * System must validate and handle errors for marketing item management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “PR Staff” by Manager |
| Postconditions: | 1. Success: Staff view marketing items successfully.  2. Fail: Staff fail in viewing marketing item. |
| Main Flow: | 1. Staff clicks on the "Manage Marketing Item" tab.  2. Displays a list of existing marketing item and management options. |
| Alternative Flows: | N/A |
| Exception: | N/A |

#### Disable Marketing Item

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Disable Marketing Item |
| Created By: | MinhHN |
| Date Created: | 21/10/2023 |
| Description: | Staff can disable marketing item. |
| Primary Actor: | Staff |
| Trigger: | The user clicks the “Disable” icon on a row. |
| Business rule: | * Ensuring marketing item information is securely stored and accessible only to authorized staff. * System must validate and handle errors for marketing item management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “PR Staff” by Manager. |
| Postconditions: | 1. Success: Staff disable marketing item successfully.  2. Fail: Staff fail in disabling marketing item |
| Main Flow: | 1. Staff clicks the “Disable” icon on a row.  2. System shows message that the marketing item is deactivated, and services are also deactivated. |
| Alternative Flows: | N/A |
| Exception: | N/A |

### <Staff> Manage Student

#### Import/Export Student File

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Import/Export Student File |
| Created By: | MinhHN |
| Date Created: | 21/10/2023 |
| Description: | Staff can import student list from a file into the system or export student list in the system to a file. |
| Primary Actor: | Staff |
| Trigger: | The user clicks the “Import Student List” button. |
| Business rule: | * Ensuring student information is securely stored and accessible only to authorized staff. * System must validate and handle errors for student management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “Student Service Staff” by Manager. |
| Postconditions: | 1. Success: Staff add Student list successfully.  2. Fail: Staff fail in adding Student. |
| Main Flow: | 1. Staff clicks the “Import Student List” button.  2. Show a pop-up for user to upload a file.  3. Staff selects a file from local file to submit.  4. System checks the validation of file type and data before uploading the data into the system.  5. System shows message that the list of students is added to the system successfully. |
| Alternative Flows: | 3a. Staff drag and drop a file into the pop-up.   1. System continues with step 4 in the Main Scenario.   1a. Staff clicks the “Export Student List” button.   1. Show a pop-up for user to choose a file type to export. 2. Staff selects a file type to proceed. 3. System shows message that the list of students is exported successfully. |
| Exception: | 1a. User upload invalid file type or invalid data validation.   1. Exception System displays error message with details of the invalid error.   2a. Student existed   1. System displays error message indicating existed Student. |

#### Update Student

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Update Student |
| Created By: | MinhHN |
| Date Created: | 21/10/2023 |
| Description: | Staff can update existing student. |
| Primary Actor: | Staff |
| Trigger: | The user clicks the “Update” icon on a row. |
| Business rule: | * Ensuring student information is securely stored and accessible only to authorized staff. * System must validate and handle errors for student management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “Student Service Staff” by Manager. |
| Postconditions: | 1. Success: Staff update Student successfully.  2. Fail: Staff fail in updating Student. |
| Main Flow: | 1. Staff clicks the “Update” icon  2. Show form to edit student information  3. Enter the new details of the student  4. System shows message that the student is edited successfully. |
| Alternative Flows: | N/A |
| Exception | 1a. User input invalid fields.   1. System displays error message indicating required re-entering invalid fields.   2a. Student existed   1. System displays error message indicating entering another Student. |

#### View Student List

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Student List |
| Created By: | MinhHN |
| Date Created: | 21/10/2023 |
| Description: | Staff can view list of existing students |
| Primary Actor: | Staff |
| Trigger: | The user clicks on the "Manage Student" tab. |
| Business rule: | * Ensuring student information is securely stored and accessible only to authorized staff. * System must validate and handle errors for student management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “Student Service Staff” by Manager. |
| Postconditions: | 1. Success: Staff view student successfully.  2. Fail: Staff fail in viewing Student. |
| Main Flow: | 1. Staff clicks on the "Manage Student" tab.  2. Displays a list of existing student and management options. |
| Alternative Flows: | N/A |
| Exception: | N/A |

#### View Student Details

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Student Details |
| Created By: | MinhHN |
| Date Created: | 23/10/2023 |
| Description: | Staff can view details of existing Student. |
| Primary Actor: | Staff |
| Trigger: | The user clicks on the "View Details" icon on a row. |
| Business rule: | * Ensuring student information is securely stored and accessible only to authorized staff. * System must validate and handle errors for student management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “Student Service Staff” by Manager. |
| Postconditions: | 1. Success: Staff view student details successfully.  2. Fail: Staff fail in viewing student details. |
| Main Flow: | 1. Staff clicks on the "View Details" icon on a row.  2. Displays a list of existing student and management options. |
| Alternative Flows: | N/A |
| Exception: | N/A |

#### Disable Students

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Disable Student |
| Created By: | MinhHN |
| Date Created: | 23/10/2023 |
| Description: | Staff can disable student. |
| Primary Actor: | Staff |
| Trigger: | The user clicks the “Disable” icon on a row. |
| Business rule: | * Ensuring student information is securely stored and accessible only to authorized staff. * System must validate and handle errors for student management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Staff”.  The Staff is assigned as “Student Service Staff” by Manager. |
| Postconditions: | 1. Success: Staff disable student successfully.  2. Fail: Staff fail in disabling student. |
| Main Flow: | 1. Staff clicks the “Disable” icon on a row.  2. System shows message that the student is deactivated, and services are also deactivated. |
| Alternative Flows: | N/A |
| Exception: | N/A |

### <Manager> Generate Report

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Generate Report |
| Created By: | MinhHN |
| Date Created: | 24/10/2023 |
| Description: | Manager can export department statistic in the system to a file. |
| Primary Actor: | Staff |
| Trigger: | The user clicks the “Generate Report” button. |
| Business rule: | * Ensuring statistical information is securely stored and accessible only to authorized manager. * System must validate and handle errors for statistical data. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Manager”. |
| Postconditions: | 1. Success: Manager export report successfully.  2. Fail: Manager fail in exporting report data. |
| Main Flow: | 1. Manager clicks the “Generate Report” button.  2. Show a pop-up for user to choose a file type to export.  3. Manager selects a file type to proceed.  4. System shows message that the list of students is exported successfully. |
| Alternative Flows: | N/A |
| Exception: | 1a. User upload invalid file type or invalid data validation.   1. System displays error message with details of the invalid error.   2a. Student existed   1. System displays error message indicating existed Student. |

### <Manager> Manage Major

#### Create Major

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Create Major |
| Created By: | MinhHN |
| Date Created: | 24/10/2023 |
| Description: | Manager can create new majors. |
| Primary Actor: | Manager |
| Trigger: | The user clicks the “Create Major” button. |
| Business rule: | * Ensuring major information is securely stored and accessible only to authorized Manager. * System must validate and handle errors for major management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Manager”. |
| Postconditions: | 1. Success: Manager creates major successfully.  2. Fail: Manag Manager fail in creating major. |
| Main Flow: | 1. Clicks on the "Create Major" button.  2. Displays a list of existing Majors  3. Selects the option to add a new major  4. Show form to enter new major details  5. Enter the details of the new major, including name and description.  6. System shows message that the majors creating successfully. |
| Alternative Flows: | N/A |
| Exception: | 1a. User input invalid fields.   1. System displays error message indicating required re-entering invalid fields.   2a. Major existed   1. System displays error message indicating entering another major. |

#### Update Major

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Update Major |
| Created By: | MinhHN |
| Date Created: | 24/10/2023 |
| Description: | Manager can update new majors. |
| Primary Actor: | Manager |
| Trigger: | The user clicks the “Update” icon on a row |
| Business rule: | * Ensuring major information is securely stored and accessible only to authorized Manager. * System must validate and handle errors for major management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Manager”. |
| Postconditions: | 1. Success: Manager update major successfully.  2. Fail: Manager fail in updating major. |
| Main Flow: | 1. Manager clicks the “Update” icon  2. Show form to edit major information  3. Enter the new details of the major  4. System shows message that the major is edited successfully. |
| Alternative Flows: | N/A |
| Exception: | 1a. User input invalid fields.   1. System displays error message indicating required re-entering invalid fields.   2a. Major existed   1. System displays error message indicating entering another major. |

#### View Majors

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Majors |
| Created By: | MinhHN |
| Date Created: | 24/10/2023 |
| Description: | Manager can view list of existing majors. |
| Primary Actor: | Manager |
| Trigger: | The user clicks on the "Manage Major" tab. |
| Business rule: | * Ensuring major information is securely stored and accessible only to authorized Manager. * System must validate and handle errors for major management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Manager”. |
| Postconditions: | 1. Success: Manager view majors successfully.  2. Fail: Manager fail in viewing major. |
| Main Flow: | 1. Manager clicks on the "Manage Major" tab.  2. Displays a list of existing major and management options. |
| Alternative Flows: | N/A |
| Exception: | N/A |

#### Disable Major

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Disable Major |
| Created By: | MinhHN |
| Date Created: | 25/10/2023 |
| Description: | Manager can disable major. |
| Primary Actor: | Manager |
| Trigger: | The user clicks the “Disable” icon on a row |
| Business rule: | * Ensuring major information is securely stored and accessible only to authorized Manager. * System must validate and handle errors for major management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Manager”. |
| Postconditions: | 1. Success: Manager disable major successfully.  2. Fail: Manager fail in disabling major. |
| Main Flow: | 1. Manager clicks the “Disable” icon on a row.  2. System shows message that the major is deactivated, and services are also deactivated. |
| Alternative Flows: | N/A |
| Exception: | N/A |

### <Manager> Manage Curriculum

#### Import/Export Curriculum File

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Import/Export Curriculum File |
| Created By: | MinhHN |
| Date Created: | 25/10/2023 |
| Description: | Manager can import curriculum list from a file into the system or export curriculum list in the system to a file. |
| Primary Actor: | Manager |
| Trigger: | The user clicks the “Import Curriculum List” button. |
| Business rule: | * Ensuring curriculum information is securely stored and accessible only to authorized staff. * System must validate and handle errors for curriculum management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Manager”. |
| Postconditions: | 1. Success: Manager adds Curriculum list successfully.  2. Fail: Manager fail in adding Curriculum. |
| Main Flow: | 1. Manager clicks the “Import Curriculum List” button.  2. Show a pop-up for user to upload a file.  3. Manager selects a file from local file to submit.  4. System checks the validation of file type and data before uploading the data into the system.  5. System shows message that the list of curriculums is added to the system successfully. |
| Alternative Flows: | 3a. Manager drag and drop a file into the pop-up.   1. System continues with step 4 in the Main Scenario.   1a. Manager clicks the “Export Curriculum List” button.   1. Show a pop-up for user to choose a file type to export. 2. Manager selects a file type to proceed. 3. System shows message that the list of curriculums is exported successfully. |
| Exception: | 1a. User upload invalid file type or invalid data validation.  1. System displays error message with details of the invalid error.  2a. Curriculum existed  1. System displays error message indicating existed Curriculum. |

#### Update Curriculum File

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Update Curriculum |
| Created By: | MinhHN |
| Date Created: | 26/10/2023 |
| Description: | Manager can update existing curriculum. |
| Primary Actor: | Manager |
| Trigger: | The user clicks the “Update” icon on a row. |
| Business rule: | * Ensuring curriculum information is securely stored and accessible only to authorized staff. * System must validate and handle errors for curriculum management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Manager”. |
| Postconditions: | 1. Success: Manager update Curriculum successfully.  2. Fail: Manager fail in updating Curriculum. |
| Main Flow: | 1. Manager clicks the “Update” icon  2. Show form to edit curriculum information  3. Enter the new details of the curriculum  4. System shows message that the curriculum is edited successfully. |
| Alternative Flows: | N/A |
| Exception: | 1a. User input invalid fields  1. System displays error message indicating required re-entering invalid fields.  2a. Curriculum existed  1. System displays error message indicating entering another Curriculum. |

#### Disable Curriculum

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Disable Curriculum |
| Created By: | MinhHN |
| Date Created: | 26/10/2023 |
| Description: | Manager can disable curriculum. |
| Primary Actor: | Manager |
| Trigger: | The user clicks the “Disable” icon on a row |
| Business rule: | * Ensuring curriculum information is securely stored and accessible only to authorized staff. * System must validate and handle errors for curriculum management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Manager”. |
| Postconditions: | 1. Success: Manager disable curriculum successfully.  2. Fail: Manager fail in disabling curriculum. |
| Main Flow: | 1. Manager clicks the “Disable” icon on a row.  2. System shows message that the curriculum is deactivated, and services are also deactivated. |
| Alternative Flows: | N/A |
| Exception: | N/A |

#### View Curriculum List

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Curriculum List |
| Created By: | MinhHN |
| Date Created: | 26/10/2023 |
| Description: | Manager can view list of existing curriculums. |
| Primary Actor: | Manager |
| Trigger: | The user clicks on the "Manage Curriculum" tab. |
| Business rule: | * Ensuring curriculum information is securely stored and accessible only to authorized staff. * System must validate and handle errors for curriculum management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Manager”. |
| Postconditions: | 1. Success: Manager view curriculum successfully.  2. Fail: Manager fail in viewing Curriculum. |
| Main Flow: | 1. Manager clicks on the "Manage Curriculum" tab.  2. Displays a list of existing curriculum and management options. |
| Alternative Flows: | N/A |
| Exception: | N/A |

#### View Curriculum Details

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Curriculum Details |
| Created By: | MinhHN |
| Date Created: | 27/10/2023 |
| Description: | Manager can view details of existing Curriculum. |
| Primary Actor: | Manager |
| Trigger: | The user clicks on the "View Details" icon on a row. |
| Business rule: | * Ensuring curriculum information is securely stored and accessible only to authorized staff. * System must validate and handle errors for curriculum management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Manager”. |
| Postconditions: | 1. Success: Manager view curriculum details successfully.  2. Fail: Manager fail in viewing curriculum details. |
| Main Flow: | 1. Manager clicks on the "View Details" icon on a row.  2. Displays Curriculum Details page with all information of a curriculum. |
| Alternative Flows: | N/A |
| Exception: | N/A |

### <Manager> Manage Staff

#### Update Staff

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | Update Staff |
| Created By: | MinhHN |
| Date Created: | 27/10/2023 |
| Description: | Manager can update existing staff. |
| Primary Actor: | Manager |
| Trigger: | The user clicks the “Update” icon on a row. |
| Business rule: | * Ensuring staff information is securely stored and accessible only to authorized staff. * System must validate and handle errors for staff management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Manager”. |
| Postconditions: | 1. Success: Manager update Staff successfully.  2. Fail: Manager fail in updating Staff. |
| Main Flow: | 1. Manager clicks the “Update” icon  2. Show form to edit staff information  3. Enter the new details of the staff.  4. System shows message that the staff is edited successfully. |
| Alternative Flows: | 3a. Manager assigns Staff with a role, each staff role includes Recruit Staff, PR Staff, Marketing Staff, and Student Service Staff.   1. System shows message that the staff is assigned successfully. |
| Exception: | 1a. User input invalid fields   1. System displays error message indicating required re-entering invalid fields.   2a. Staff existed   1. System displays error message indicating entering another Staff. |

#### View Staff List

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Staff List |
| Created By: | MinhHN |
| Date Created: | 27/10/2023 |
| Description: | Manager can view list of existing staff. |
| Primary Actor: | Manager |
| Trigger: | The user clicks on the "Manage Staff" tab. |
| Business rule: | * Ensuring staff information is securely stored and accessible only to authorized staff. * System must validate and handle errors for staff management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Manager”. |
| Postconditions: | 1. Success: Manager view staff successfully.  2. Fail: Manager fail in viewing Staff. |
| Main Flow: | 1. Manager clicks on the "Manage Staff" tab.  2. Displays a list of existing staff and management options. |
| Alternative Flows: | N/A |
| Exception: | N/A |

#### View Staff Details

a, GUI

b, Use Case Specification

|  |  |
| --- | --- |
| Use Case ID: |  |
| Use Case Name: | View Staff Details |
| Created By: | MinhHN |
| Date Created: | 27/10/2023 |
| Description: | Manager can view details of existing Staff. |
| Primary Actor: | Manager |
| Trigger: | The user clicks on the "View Details" icon on a row. |
| Business rule: | * Ensuring staff information is securely stored and accessible only to authorized staff. * System must validate and handle errors for staff management. |
| Include use cases: | N/A |
| Extends use cases: | N/A |
| Preconditions: | The user is logged into the system as “Manager”. |
| Postconditions: | 1. Success: Manager view staff details successfully.  2. Fail: Manager fail in viewing staff details |
| Main Flow: | 1. Manager clicks on the "View Details" icon on a row.  2. Displays Staff Details page with all information of a staff. |
| Alternative Flows: | N/A |
| Exception: | N/A |

# Data Requirements

## Logical Data Model

|  |  |
| --- | --- |
| Table | Description |
| Application | Primary key: application\_id  Foreign key:   * employee\_id * student\_id |
| Appointment\_Schedule | Primary key: appointment\_id  Foreign key:   * application\_id * student\_id |
| Campaign | Primary key: campaign\_id  Foreign key: application\_id |
| Campaign\_Staff | Primary key:   * campaign\_id * employee\_id |
| Communication | Primary key: communication\_id  Foreign key:   * employee\_id * student\_id * applicant\_id |
| Contract | Primary key: \_id  Foreign key:   * financial\_id * student\_id |
| Course | Primary key: course\_id  Foreign key:   * application\_id * major\_id |
| Employee | Primary key: employee\_id  Foreign key: role\_id |
| Event | Primary key: event\_id  Foreign key: application\_id |
| Financial | Primary key: financial\_id  Foreign key: application\_id |
| Major | Primary key: major\_id |
| Nationality | Primary key: nationality\_id |
| Ranking | Primary key: \_id  Foreign key: major\_id |
| Report | Primary key: report\_id  Foreign key: employee\_id |
| Role | Primary key: role\_id |
| Student | Primary key: student\_id |
| Student\_Nationality | Primary key:   * nationality\_id * student\_id |
| System Setting | Primary key: settings\_id  Foreign key: employee\_id |
| Task | Primary key: task\_id  Foreign key: employee\_id |
| Tuition | Primary key: \_id  Foreign key: major\_id |

## Data Dictionary

### Application

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| application\_id | integer |  | x |  | PK |
| employee\_id | integer |  |  |  | FK |
| student\_id | integer |  |  |  | FK |
| application\_submitted\_documents | varchar | 750 |  |  |  |
| application\_test\_result | varchar | 155 |  |  |  |
| application\_other\_information | varchar | 750 |  | x |  |
| application\_grades | float |  |  | x |  |
| application\_status | boolean |  |  |  |  |

### Appointment\_Schedule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| appointment\_id | integer |  | x |  | PK |
| application\_id | integer |  |  |  | FK |
| student\_id | integer |  |  |  | FK |
| appointment\_datetime | datetime |  |  |  |  |
| appointment\_location | varchar | 255 |  |  |  |
| appointment\_participants | varchar | 155 |  |  |  |

### Campaign

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| campaign\_id | integer |  | x |  | PK |
| application\_id | integer |  |  |  | FK |
| campaign\_name | varchar | 155 |  |  |  |
| campaign\_type | varchar | 50 |  |  |  |
| campaign\_result | varchar | 155 |  | x |  |

### Campaign\_Staff

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| campaign\_id | integer |  |  |  | PK |
| application\_id | integer |  |  |  | PK |

### Communication

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| communication\_id | integer |  | x |  | PK |
| employee\_id | integer |  |  |  | FK |
| student\_id | integer |  |  |  | FK |
| application\_id | integer |  |  |  | FK |
| communication\_datetime | datetime |  |  |  |  |
| communication\_type | varchar | 50 |  |  |  |
| communication\_content | varchar | 750 |  | x |  |

### Contract

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| contract\_id | integer |  | x |  | PK |
| student\_id | integer |  |  |  | FK |
| financial\_id | integer |  |  |  | FK |
| contract\_type | varchar | 155 |  |  |  |
| contract\_date | datetime |  |  |  |  |
| contract\_terms | varchar | 255 |  | x |  |

### Course

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| course\_id | integer |  | x |  | PK |
| application\_id | integer |  |  |  | FK |
| major\_id | integer |  |  |  | FK |
| course\_name | varchar | 155 |  |  |  |
| course\_type | varchar | 50 |  |  |  |
| course\_result | varchar | 155 |  | x |  |

### Employee

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| employee\_id | integer |  | x |  | PK |
| role\_id | integer |  |  |  | FK |
| employee\_name | varchar | 155 |  |  |  |
| employee\_position | varchar | 50 |  |  |  |
| employee\_department | varchar | 155 |  |  |  |

### Event

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| event\_id | integer |  | x |  | PK |
| application\_id | integer |  |  |  | FK |
| event\_name | varchar | 155 |  |  |  |
| event\_date | datetime |  |  |  |  |
| event\_location | varchar | 255 |  |  |  |
| event\_registration\_status | boolean |  |  |  |  |

### Financial

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| financial\_id | integer |  | x |  | PK |
| application\_id | integer |  |  |  | FK |
| financial\_payment\_status | varchar | 155 |  |  |  |
| financial\_contract\_management | varchar | 155 |  |  |  |

### Major

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| major\_id | integer |  | x |  | PK |
| major\_name | varchar | 50 |  |  |  |

### Nationality

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| nationality\_id | integer |  | x |  | PK |
| nationality\_name | varchar | 155 |  |  |  |

### Ranking

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| ranking\_id | integer |  | x |  | PK |
| major\_id | integer |  |  |  | FK |
| ranking\_score | integer |  |  |  |  |
| ranking\_name | varchar | 155 |  |  |  |

### Report

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| report\_id | integer |  | x |  | PK |
| employee\_id | integer |  |  |  | FK |
| report\_name | varchar | 155 |  |  |  |
| report\_content | varchar | 750 |  |  |  |

### Role

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| role\_id | integer |  | x |  | PK |
| role\_name | varchar | 50 |  |  |  |
| role\_permissions | varchar | 50 |  |  |  |

### Student

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| student\_id | integer |  | x |  | PK |
| student\_name | varchar | 155 |  |  |  |
| student\_address | varchar | 255 |  | x |  |
| student\_contact\_info | varchar | 255 |  | x |  |
| student\_education\_info | varchar | 255 |  | x |  |
| student\_skill | varchar | 255 |  | x |  |
| student\_interests | varchar | 255 |  | x |  |

### Student\_Nationality

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| nationality\_id | integer |  |  |  | PK |
| student\_id | integer |  |  |  | PK |

### System\_Settings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| settings\_id | integer |  | x |  | PK |
| employee\_id | integer |  |  |  | FK |
| settings\_configuration | varchar | 750 |  |  |  |

### Task

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| task\_id | integer |  | x |  | PK |
| employee\_id | integer |  |  |  | FK |
| task\_name | varchar | 155 |  |  |  |
| task\_description | varchar | 750 |  |  |  |
| task\_due\_date | datetime |  |  |  |  |

### Tuition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Unique** | **Null** | **PK/FK** |
| tuition\_id | integer |  | x |  | PK |
| major\_id | integer |  |  |  | FK |
| tuition\_name | float |  |  |  |  |

## Reports

* **Admissions Summary Report**: This report will provide a summary of the admissions process, including the number of inquiries, applications, and admissions decisions.
* **Applicant Pipeline Report**: This report will show the status of each applicant in the admissions process.
* **Admissions Funnel Report**: This report will show the number of applicants at each stage of the admissions process.

## Data Acquisition, Integrity, Retention, and Disposal

* **Data validation**: The SASS will validate all data entered by users to ensure that it is accurate and complete.
* **Backups**: The SASS will be backed up regularly to protect against data loss.
* **Checkpointing**: The SASS will use checkpoints to ensure that data is not lost in the event of a system failure.

# External Interface Requirements

## User Interfaces

* **Usability**: The user interfaces will be easy to use and navigate.
* **Accessibility**: The user interfaces will be accessible to users with disabilities.
* **Security**: The user interfaces will be secure and protect the privacy of user data.

## Software Interfaces

* **Student information system:** The SASS will exchange data with the student information system, such as student names and contact information.
* **Website**: The SASS will integrate with the FPT University website to allow prospective students to submit online applications and track their application status.

## Hardware Interfaces

*N/A*

## Communications Interfaces

* **Email**: The SASS will use email to send notifications to users, such as application status updates and admissions decisions.
* **Web browser**: The SASS will be accessed by users through a web browser.

# Quality Attributes

## Usability

* The user interface will be simple and intuitive.
* The user interface will be consistent across all pages and screens.
* The user interface will be accessible to users with disabilities.
* The user interface will be responsive and load quickly.
* The user interface will provide clear and concise error messages.
* The user interface will provide help documentation and tutorials.

## *Performance*

* The SASS will be able to handle 100 concurrent users without any noticeable performance degradation.
* The SASS will be able to load a web page in less than 2 seconds.
* The SASS will be able to process an application in less than 5 seconds.

## Security

* All user data will be encrypted at rest and in transit.
* Only authorized users will have access to user data.
* The SASS will be regularly scanned for vulnerabilities.
* The SASS will be deployed on a secure server infrastructure.

## Safety

* The SASS will prevent users from submitting incomplete or inaccurate applications.
* The SASS will prevent users from viewing or modifying other users' data.
* The SASS will prevent unauthorized access to the system.
* The SASS will prevent data loss or corruption.

# Internationalization and Localization Requirements

The SASS must be internationalized and localized to support users in different countries and regions. The following requirements must be met:

* The SASS must be able to display text in multiple languages.
* The SASS must be able to format dates, numbers, addresses, and telephone numbers in accordance with local conventions.
* The SASS must be able to use the correct currency symbols and currency conversion rates for different countries.
* The SASS must be able to support different time zones.
* The SASS must comply with all applicable international regulations and laws.

# Other Requirements

* **Legal and Regulatory Compliance:** The CRM system must comply with relevant legal and regulatory requirements, including data privacy and protection laws.
* **Installation and Configuration:** The system should provide easy installation and configuration procedures for administrators.
* **Logging and Monitoring:** The system should have comprehensive logging, monitoring, and audit trail capabilities to track system activities and security events.

**Appendix A: Glossary**

* Data flow diagram
* Feature tree
* State-transition diagram

The data flow diagram shows the flow of data through the SASS. The feature tree shows the features of the SASS and their relationships to each other. The state-transition diagram shows the different states of the SASS and the transitions between those states.

**Appendix B: Analysis Models**

* The SASS should be designed using a Unicode character set to support multiple languages.
* The SASS should use a translation management system to streamline the translation process.
* The SASS should be tested with users in different countries and regions to ensure that it is localized correctly.