**FPT ACADEMY INTERNATIONAL**



**College-Website**



|  |  |  |
| --- | --- | --- |
| **Center Name:** | FPT – Aptech – HCM | |
|  | Address:590 Cach Mang Thang Tam,  District 3, Ho Chi Minh City | |
| **Instructor:** | Mr. Ngo Phuoc Nguyen | |
| **Class:** | T1.2009.A0 | |
| **Group:** | 01 | |
| **Group members** | Name  Luu Thanh Sang | Student ID  Student1278914 | |
|  | Nguyen Quang Nguyen | Student1279066 | |
|  | Luu Huy Hoang  Nguyen Duc Nam | Student1279605  Student1279119 | |
|  |  |  | |
|  |

**January-2022**

**TABLE OF CONTENTS:**

This is to certify that

Mr. Luu Thanh Sang

Mr. Nguyen Quang Nguyen

Mr. Luu Huy Hoang

Mr. Nguyen Duc Nam



have successfully designed and

developed ****

Submitted by:

Date of Issue: \_\_\_\_\_\_\_\_

Authorized Signature: \_\_\_\_\_\_\_\_\_\_\_



[REVIEW I 6](#_Toc94364454)

[Acknowledgment 7](#_Toc94364455)

[Problem Definition 8](#_Toc94364456)

[1. INTRODUCTION 8](#_Toc94364457)

[2. PROJECT REQUIREMENT 8](#_Toc94364458)

[3. PROJECT DESIGNING 8](#_Toc94364459)

[4. Functional Logic 9](#_Toc94364460)

[ADMISSION PROCEDURE: 9](#_Toc94364461)

[Hardware / Software Requirements 11](#_Toc94364462)

[1. Server Requirement 11](#_Toc94364463)

[2. Client Requirement 12](#_Toc94364464)

[Task Sheet Review I 13](#_Toc94364465)

[REVIEW II 14](#_Toc94364466)

[1. USE CASES 15](#_Toc94364467)

[HOME PAGE 16](#_Toc94364468)

[COURSE 16](#_Toc94364469)

[ADMISSION 17](#_Toc94364470)

[ADMISSION FORM 17](#_Toc94364471)

[CHECK THE ADMISSION STATUS 19](#_Toc94364472)

[DEPARTMENTS 20](#_Toc94364473)

[FACULTY 20](#_Toc94364474)

[FACILITIES 21](#_Toc94364475)

[STUDENT REGISTRATION 21](#_Toc94364476)

[CONTACT US 23](#_Toc94364477)

[FEEDBACK 23](#_Toc94364478)

[CREATE/EDIT COURSE 26](#_Toc94364479)

[DELETE COURSE 27](#_Toc94364480)

[EDIT ADMISSION 28](#_Toc94364481)

[CREATE/EDIT DEPARTMENT 28](#_Toc94364482)

[DELETE DEPARTMENT 29](#_Toc94364483)

[CREATE/EDIT FACULTY 30](#_Toc94364484)

[DELETE FACULTY 31](#_Toc94364485)

[CREATE/EDIT FACILITIES 32](#_Toc94364486)

[DELETE FACILITIES 33](#_Toc94364487)

[CREATE/EDIT STREAM 34](#_Toc94364488)

[DELETE STREAM 34](#_Toc94364489)

[CREATE/EDIT FIELD 35](#_Toc94364490)

[DELETE FIELD 36](#_Toc94364491)

[CREATE/EDIT SPECIALIZED SUBJECT 37](#_Toc94364492)

[DELETE SPECIALIZED SUBJECT 38](#_Toc94364493)

[CREATE/EDIT OPTIONAL SUBJECT 39](#_Toc94364494)

[DELETE OPTIONAL SUBJECT 40](#_Toc94364495)

[CREATE/EDIT ACCOUNT 40](#_Toc94364496)

[DELETE ACCOINT 42](#_Toc94364497)

[LOGIN 43](#_Toc94364498)

[LOG OUT 44](#_Toc94364499)

[2. SEQUENCE DIAGRAMS 45](#_Toc94364500)

[ADMISSION 45](#_Toc94364501)

[CHECK ADMISSION STATUS 45](#_Toc94364502)

[STUDENT REGISTRATION 46](#_Toc94364503)

[ADMIN LOGIN 47](#_Toc94364504)

[ADMIN LOGOUT 47](#_Toc94364505)

[ADMIN SYSTEM MANAGEMENT 48](#_Toc94364506)

[CREATE/EDIT ADMIN ACCOUNT 49](#_Toc94364507)

[3. DATA FLOW DIAGRAM 50](#_Toc94364508)

[CONTEXT DIAGRAM 50](#_Toc94364509)

[LEVEL 0 50](#_Toc94364510)

[LEVEL 1 52](#_Toc94364511)

[4. LOGICAL ENTITY RELATIONSHIP DIAGRAM 57](#_Toc94364512)

[5. ENTITY PROPERTIES 58](#_Toc94364513)

[STREAM 58](#_Toc94364514)

[FIELD 58](#_Toc94364515)

[COURSE 58](#_Toc94364516)

[ADMISSION 59](#_Toc94364517)

[SPECIALIZED SUBJECT 59](#_Toc94364518)

[OPTIONAL SUBJECT 60](#_Toc94364519)

[REGISTRATION 60](#_Toc94364520)

[DEPARTMENT 60](#_Toc94364521)

[FACULTY 61](#_Toc94364522)

[FACILITIES 61](#_Toc94364523)

[FEEDBACK 62](#_Toc94364524)

[ACCOUNT 62](#_Toc94364525)

[6. DATABASE DIAGRAM 63](#_Toc94364526)

[7. TABLE DESIGN 64](#_Toc94364527)

[STREAM 64](#_Toc94364528)

[FIELD 64](#_Toc94364529)

[COURSE 64](#_Toc94364530)

[ADMISSION 65](#_Toc94364531)

[SPECIALZED SUBJECT 66](#_Toc94364532)

[OPTINAL SUBJECT 66](#_Toc94364533)

[REGISTRATION 66](#_Toc94364534)

[DEPARTMENT 67](#_Toc94364535)

[FACULTY 67](#_Toc94364536)

[FACILITIES 67](#_Toc94364537)

[FEEDBACK 68](#_Toc94364538)

[ACCOUNT 68](#_Toc94364539)

# REVIEW I

## Acknowledgment

We would like to acknowledge all those who have given support and help us make the project a success.

We wish to express our deep gratitude to all teachers who have been devoting their lives to teach us how to stand-alone and walk ahead.

We are grateful to our families who take care and encourage us even though we are successful or failed, also to our friends who are always care of us. They never leave us alone and always look forward to us when we are on any road of the life.

We are much thankful to the entire staff and chairpersons at the Head Office of Aptech Worldwide, Aptech – Vietnam, and FPT – Aptech Center who have been organizing and looking after our studying course.

There are no words to show our appreciation for teachers of FPT Aptech Centre who have been organizing and looking after us during our studying course to finish this project. Our special thanks Mr Tran Phuoc Sinh who works day by day to teach and guide us, e-Project Team at the Head Office who instruct and help us.

Finally, we would like to offer many thanks to all our schoolfellows for their valuable suggestions.

We would like to thank sincerely!

## Problem Definition

### INTRODUCTION

As of today world all the manual paper work like accounting, maintaining records are computerized. Even the college and school admission, result declaration is performed online today. Due to which all the school and colleges felt the need of making the website of their organization. The main purpose of making website is to attract the masses and advertisement of the school, colleges, and their achievements to the people leaving in any area (near or remote). The **ITM** College wants the website of the college to be developed and they have approached you for the same.

### PROJECT REQUIREMENT

The College had provided all the content details they want to add in the website. The details of the information and what link the Home page must contain? Name of the links required in the website for navigation. There should be section in the Home Page for Upcoming events in the college, Achievements, New course, Merit list. The Link like **HOME, COURSES, STUDENT REGISTRATION, DEPARTMENTS, FACULTY, FACILITIES, ONLINE ADMISSION, CONTACT US, FEEDBACK, ADMIN LOGIN.**

### PROJECT DESIGNING

**HOME PAGE:** Here there should be an introduction of the college with hyperlink at the top (for navigation) and various sections to highlight the important features of the college. There should be separate section for the Admin Login (where the details of website can be modified and updated).

**COURSES:** Here the details of all the courses must be stated which are available in the college.

**ADMISSION:** Here there will admission form for the students. All the admission form related to the particular subject must be there. Students can download the form or can also fill the form online and submit. Once the online form is submitted then the student will get a unique registration number and this number must be used to check the admission status. If the student status is accepted then they can proceed the online submission *by clicking proceed* button or link and complete the further details like selection of subjects.

**DEPARTMENTS:** Here the details of the entire department must be added (like chemistry, French, Economics, Psychology, Geology and Mathematics).

**FACULTY:** Here it should state the details of the entire faculty (should be categorized according to their departments).

**FACILITIES:** Like what all Facilities are provided in the college such as college canteen, hostel, placement center, gymkhana, college library, administrator office)

**STUDENT REGISTRATION:** here the student can register themselves with the help of their ID (unique) so that they can get the updates of their colleges, assignments, exam time table. If the same student tries to register again with the same ID then an error message must be thrown.

**CONTACT US:** It should state the contact details of the college (must cover all the branches).

**FEEDBACK FORM:** Here the feedback is taken from the end user.

**ADMIN LOGIN:** There should be separate section for login. The admin person will be able to update the data of the website.

### Functional Logic

#### ADMISSION PROCEDURE:

* The Admin will have the responsibilities of checking which student is eligible for the admission.
* When an Online admission form is filling by the student they will provide all the details of the previous education. In the first level of form filling the student is supposed to provide personal details like
  + Name
  + Father Name
  + Mother Name
  + Date of Birth( as per the certificate)
  + Gender
  + Residential Address
  + Permanent Address
  + Admission for ( selecting a stream, field)
  + In tabular format ( details of previous exam)
    - University
    - Enrollment number
    - Center
    - Stream
    - Field
    - Marks secured
    - Out of
    - Class Obtained
  + Sports details (if any)
  + Submit ( for submitting the details)

Once the candidate click on the submit button a Unique number must be provided to them.

* With the help of the Unique ID they can check the status of the admission like whether waiting, accepted.
* Once the status is confirmed, they can provide further details like specialized subject, optional subject and finally can print the form for seeking admission in the college.

## Hardware / Software Requirements

### Server Requirement

* Hardware

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| **CPU** | Processor type:  Pentium IV-compatible processor or faster  Processor speed:  Recommended: 2.0 GHz or faster |
| **OS** | Microsoft Windows Server 2012 with IIS |
| **Memory (RAM)** | RAM:  Minimum: 512 MB  Recommended: 2 GB or more |
| **Hard Drive** | Free space:  Minimum: 200 MB  Recommended: 50 GB or more  Maximum: Operating system maximum |

* Software

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| **Microsoft .NET Framework** | Version 4.6 or higher |
| **RDBMS** | Microsoft SQL Server 2019 |

### Client Requirement

* Hardware

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| **CPU** | Processor type:  Pentium III-compatible processor or faster  Processor speed:  Recommended: 1.0 GHz or faster |
| **OS** | All OS (Window, Linux, Android, Mac OS …) |
| **Memory (RAM)** | RAM:  Minimum: 512 MB  Recommended: 1 GB or more  Maximum: Operating system maximum |
| **Hard Drive** | Free space:  Minimum: 10 MB |

* Software

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| **Web Browser** | Chrome, Opera (up to date) |

* Technology

|  |  |
| --- | --- |
| **Frontend** | ASP.NET MVC |
| **Backend** | ASP.NET MVC  SQL Server 2019 |

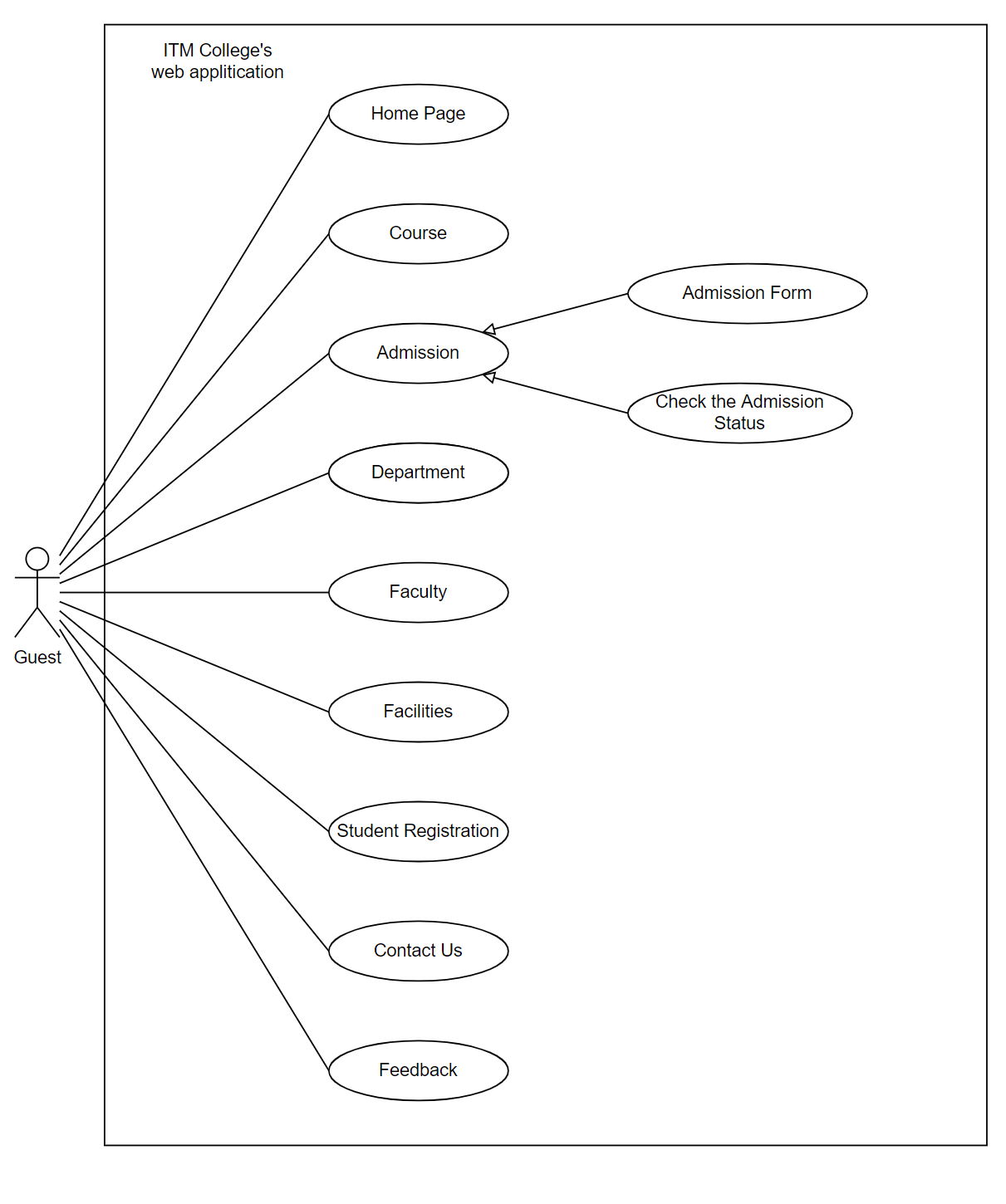
## Task Sheet Review I

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Ref. No: 4 | Project Title:  Railway Reservation Manage System | Date of Preparation of Activity Plan | | | |
| No. | Task | Actual Start Date | Actual Days | Team Member Names | Status |
| 01 | Acknowledgment | 1 Jan , 2022 | 2 | All Members | Completed |
| 02 | Problem Definition | All Members | Completed |
| 03 | Customer Requirement | All Members | Completed |
| 05 | Hardware/Software | All Members | Completed |
| 06 | Task sheet | All Members | Completed |

|  |  |  |
| --- | --- | --- |
|  | Prepare By: Group 2 | Approved By: Faculty |
| Date: Jan , 2022 | Team Leader  Luu Thanh Sang | Mr. Ngo Phuoc Nguyen |

# REVIEW II

### USE CASES



#### HOME PAGE

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.1 | |
| **Use Case Name** | Home Page | |
| **Author** | All team | |
| **Actors** | Guest | |
| **Description** | Guest view home page | |
| **Trigger** | Guest access on homepage | |
| **Pre-conditions** | The guest’s device is already connected to the internet | |
| **Post-conditions** | Success: Show homepage | |
| Fail: No page is shown | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to homepage | 2. System shows the homepage |

#### COURSE

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.2 | |
| **Use Case Name** | Course | |
| **Author** | All team | |
| **Actors** | Guest | |
| **Description** | Guest view details of all course | |
| **Trigger** | Guest want to view details of all course must be started which are available | |
| **Pre-conditions** | The guest’s device is already connected to the internet | |
| **Post-conditions** | Success: Show Course page | |
| Fail: No page is shown | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to Course page | 2. System shows the Course page |

#### ADMISSION

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.3 | |
| **Use Case Name** | Admission | |
| **Author** | All team | |
| **Actors** | Guest | |
| **Description** | Guest view admission form and check status of admission status | |
| **Trigger** | Guest want to view admission form and check status of admission status | |
| **Pre-conditions** | The guest’s device is already connected to the internet | |
| **Post-conditions** | Success: Show admission page | |
| Fail: No page is shown | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to admission page | 2. System shows the admission page |

#### ADMISSION FORM

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.3.1 | |
| **Use Case Name** | Admission form | |
| **Author** | All team | |
| **Actors** | Guest | |
| **Description** | Guest fill the admission form | |
| **Trigger** | Guest want to view admission | |
| **Pre-conditions** | The guest’s device is already connected to the internet | |
| **Post-conditions** | Success: Guest’s admission is saved | |
| Fail: Guest’s admission is not saved | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to admission page  3. Actor input the information and click on “Submit” button | 2. System shows the admission form with following controls:  - “Full Name” text field  - “Father Name” text field  - “Mother Name” text field  - “Date of Birth” text field  - “Email” text field  - “Gender” check box  - “Residential Address” text field  - “Permanent Address” text field  - “Stream” select box  - “Field” select box  - “Sports” text field  - Details of previous exam  + “University” text field  + “Enrollment number” text field  + “Center” text field  + “Stream” text field  + “Field” text field  + “Marks secured” text field  + “Out of date” text field  + “Class Obtained” text field  - “Submit” button  4. System validates the information  5. System save the information to the database |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Full Name” text field blank | Full Name cannot be blank |
| Actor left “Email” text field | Email cannot be blank |
| Actor left “Father Name” text field blank | Father Name cannot be blank |
| Actor left “Mother Name” text field blank | Mother Name cannot be blank |
| Actor left “Date Of Birth” text field blank | Date Of Birth cannot be blank |
| Actor left “Residential Address” text field blank | Residential Address cannot be blank |
| Actor left “Permanent Address” text field blank | Permanent Address cannot be blank |
| Actor left “Stream” select box | Stream cannot be blank |
| Actor left “Field” select box | Field cannot be blank |

#### CHECK THE ADMISSION STATUS

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.3.2 | |
| **Use Case Name** | Check the admission status | |
| **Author** | All team | |
| **Actors** | Guest | |
| **Description** | Guest check the admission status | |
| **Trigger** | Guest want to check the admission status | |
| **Pre-conditions** | The guest’s device is already connected to the internet | |
| **Post-conditions** | Success: Show the admission status | |
| Fail: Guest’s admission status is not shown | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to admission’s page  3. Actor input the information  4. Action click on “Submit” button | 2. System shows the check admission status form with following controls:  - “Registration Number” text field  - “Submit” button  5. System validates the information  6. System show the admission status |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Registration Number” text field blank | Registration Number cannot be blank |
| Actor input the invalid registration number | Registration Number is invalid |

#### DEPARTMENTS

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.4 | |
| **Use Case Name** | Departments | |
| **Author** | All team | |
| **Actors** | Guest | |
| **Description** | Guest view details of the entire department | |
| **Trigger** | Guest want to view details of the entire department | |
| **Pre-conditions** | The guest’s device is already connected to the internet | |
| **Post-conditions** | Success: Show department page | |
| Fail: No page is shown | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to department page | 2. System shows the department page |

#### FACULTY

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.5 | |
| **Use Case Name** | Faculty | |
| **Author** | All team | |
| **Actors** | Guest | |
| **Description** | Guest view details of the entire faculty | |
| **Trigger** | Guest want to view details of the entire faculty | |
| **Pre-conditions** | The guest’s device is already connected to the internet | |
| **Post-conditions** | Success: Show faculty page | |
| Fail: No page is shown | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to faculty page | 2. System shows the faculty page |

#### FACILITIES

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.6 | |
| **Use Case Name** | Facilities | |
| **Author** | All team | |
| **Actors** | Guest | |
| **Description** | Guest view details of all facilities are provide in the college | |
| **Trigger** | Guest want to view details of all facilities are provide in the college | |
| **Pre-conditions** | The guest’s device is already connected to the internet | |
| **Post-conditions** | Success: Show facilities page | |
| Fail: No page is shown | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to facilities page | 2. System shows the facilities page |

#### STUDENT REGISTRATION

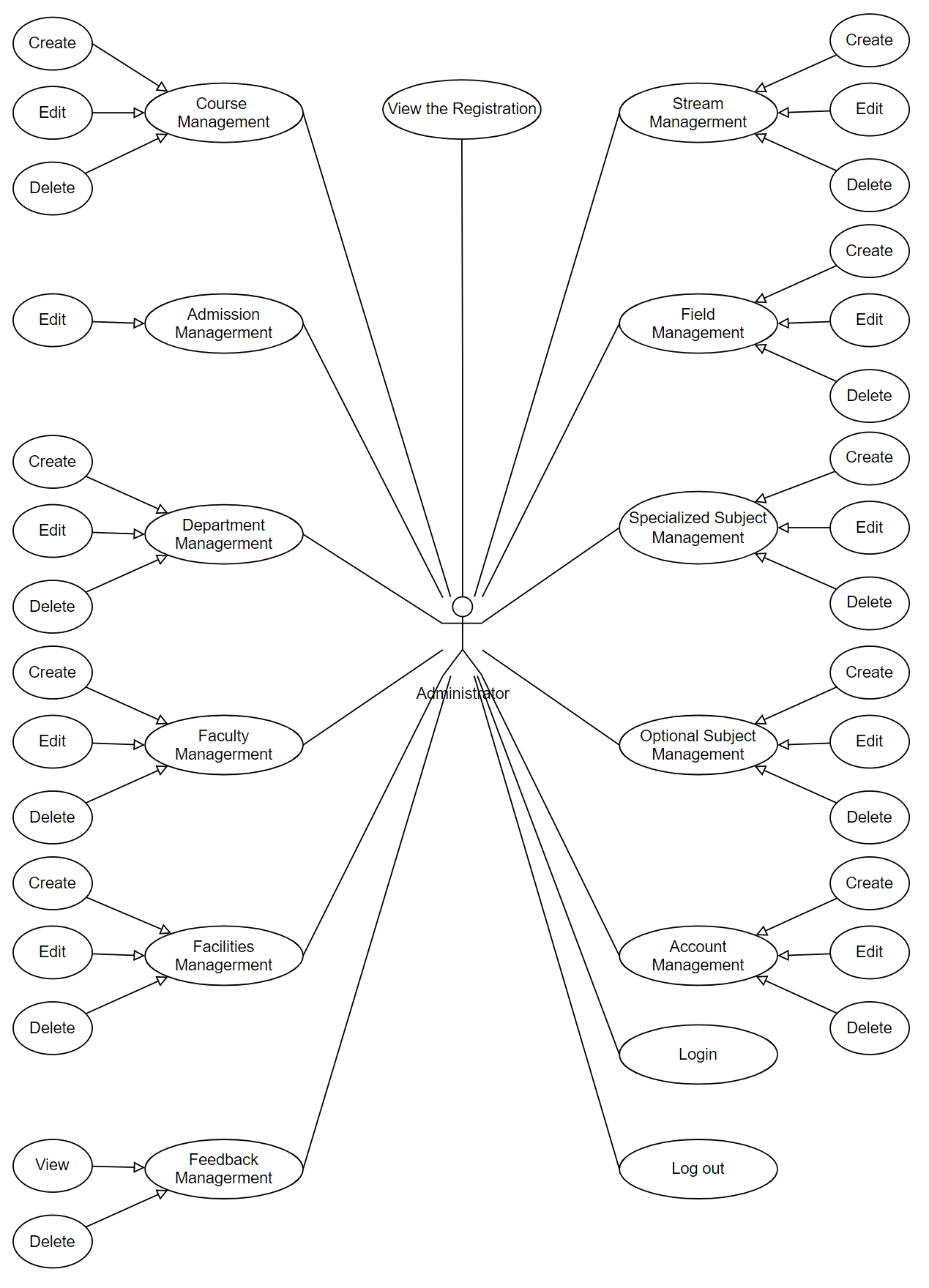
|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.7 | |
| **Use Case Name** | Student Registration | |
| **Author** | All team | |
| **Actors** | Guest | |
| **Description** | Guest view the registration form | |
| **Trigger** | Guest want to fill the registration form | |
| **Pre-conditions** | The guest’s device is already connected to the internet | |
| **Post-conditions** | Success: Show registration page | |
| Fail: No page is shown | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to student registration page  3. Actor input the information and click on “Register” button | 2. System shows the registration page with following controls:  - “Name” text field  - “Father Name” text field  - “Mother Name” text field  - “Date of Birth” text field  - “Gender” check box  - “Residential Address” text field  - “Permanent Address” text field  - “Stream” select box  - “Field” select box  - “Sports” text field  - “Admission for” select box  - Details of previous exam  + “University” text field  + “Enrollment number” text field  + “Center” text field  + “Stream” text field  + “Field” text field  + “Marks secured” text field  + “Out of date” text field  + “Class Obtained” text field  - “Specialized Subject” select box  - “Optional Subject” select box  - “Choose File” upload image button  - “Person to contact in case of emergency” text field  - “Address of person to contact in case of emergency” text field  - “Phone Number of person to contact in case of emergency” text field  - “Submit” button  4. System validates the information  5. System insert the information to database  6. System redirects to page which contain success registration message. |

#### CONTACT US

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.8 | |
| **Use Case Name** | Contact Us | |
| **Author** | All team | |
| **Actors** | Guest | |
| **Description** | Guest view the contact details of the college | |
| **Trigger** | Guest want to view the contact details of the college | |
| **Pre-conditions** | The guest’s device is already connected to the internet | |
| **Post-conditions** | Success: Show Contact Us page | |
| Fail: No page is shown | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to Contact Us page | 2. System shows the Contact Us page |

#### FEEDBACK

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.9 | |
| **Use Case Name** | Feedback | |
| **Author** | All team | |
| **Actors** | Guest | |
| **Description** | Guest view feedback form | |
| **Trigger** | Guest want to feedback | |
| **Pre-conditions** | The guest’s device is already connected to the internet | |
| **Post-conditions** | Success: Show feedback page | |
| Fail: No page is shown | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to feedback page  4. Actor fill the information and click on “Send” button | 2. System shows the feedback form with following controls:  - “First Name” text field  - “Last Name” text field  - “Email” text field  - “Subject” text field  - “Message” text field  - “Send” button  5. System validates the information  6. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “First Name” text field blank | First Name cannot be blank |
| Actor left “Last Name” text field blank | Last Name cannot be blank |
| Actor left “Email” text field blank | Email cannot be blank |
| Actor left “Subject” text field | Subject cannot be blank |
| Actor left “Message” text field | Message cannot be blank |
| Actor input an invalid email | Email is invalid |



#### CREATE/EDIT COURSE

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.1, UC-2.2 | |
| **Use Case Name** | Create/Edit Course | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin create or edit a course | |
| **Trigger** | Admin want to create/edit a course | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: A course is created/edited successfully | |
| Fail: Course is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to course management and click on Create Course link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit course form with following controls:  - “Course Name” text field  - “Description” text field  - “Stream” select box  - “Field” select box  - “Image” select file button  - “Create” / “Save” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Course Name” text field blank | Course Name cannot be blank |
| Actor left “Stream” select box | Stream cannot be blank |
| Actor left “Field” select box | Field cannot be blank |

#### DELETE COURSE

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.3 | |
| **Use Case Name** | Delete Course | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin delete a course | |
| **Trigger** | Admin want to delete a course | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: Course is deleted successfully | |
| Fail: Course is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to course management and click on Delete button  3. Actor click on “Yes” button | 2. System shows confirm message box with following controls:  - “Yes” button  - “No” button  4. System delete the correspond record in database and redirect to course management page |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor click “No” button on confirm message box | No record is deleted and system redirect to course management page |

#### EDIT ADMISSION

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.4 | |
| **Use Case Name** | Edit Admission | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin view and edit an admission | |
| **Trigger** | Admin want to view and edit an admission | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: Admission is edited successfully | |
| Fail: Admission is edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to Admission management and click on Edit button  3. Actor click on “Save” button | 2. System shows the edit admission page with following controls:  - “Status” select box  - “Save” button  4. System save the information to database and redirect to Admission Management page |

#### CREATE/EDIT DEPARTMENT

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.5, UC-2.6 | |
| **Use Case Name** | Create/Edit Department | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin create or edit a department | |
| **Trigger** | Admin want to create/edit a department | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: A department is created/edited successfully | |
| Fail: Department is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to department management and click on Create Department link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit department form with following controls:  - “Name” text field  - “Description” text field  - “Image” select file button  - “Create”/”Save” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Name” text field blank | Name cannot be blank |
| Actor input a existed department name . | Name is existed. Please choose another one |

#### DELETE DEPARTMENT

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.7 | |
| **Use Case Name** | Delete Department | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin delete a department | |
| **Trigger** | Admin want to delete a department | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: Department is deleted successfully | |
| Fail: Department is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to department management and click on Delete button  3. Actor click on “Yes” button | 2. System shows confirm message box with following controls:  - “Yes” button  - “No” button  4. System delete the correspond record in database and redirect to department management page |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor click “No” button on confirm message box | No record is deleted and system redirect to department management page |

#### CREATE/EDIT FACULTY

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.8, UC-2.9 | |
| **Use Case Name** | Create/Edit Faculty | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin create or edit a faculty | |
| **Trigger** | Admin want to create/edit a faculty | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: A faculty is created/edited successfully | |
| Fail: Faculty is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to faculty management and click on Create Faculty link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit faculty form with following controls:  - “Name” text field  - “DOB” text field  - “Degree” select box  - “Department” select box  - “Image” select file button  - ““Create” / “Save” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Name” text field blank | Name cannot be blank |
| Actor left “DOB” text field | DOB cannot be blank |
| Actor left “Degree” text field | Degree cannot be blank |
| Actor left Department Name select box | Department Name cannot be blank |

#### DELETE FACULTY

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.10 | |
| **Use Case Name** | Delete Faculty | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin delete a faculty | |
| **Trigger** | Admin want to delete a faculty | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: Faculty is deleted successfully | |
| Fail: Faculty is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to faculty management and click on Delete button  3. Actor click on “Yes” button | 2. System shows confirm message box with following controls:  - “Yes” button  - “No” button  4. System delete the correspond record in database and redirect to faculty management page |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor click “No” button on confirm message box | No record is deleted and system redirect to faculty management page |

#### CREATE/EDIT FACILITIES

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.11, UC-2.12 | |
| **Use Case Name** | Create/Edit Facilities | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin create or edit a facilities | |
| **Trigger** | Admin want to create/edit a facilities | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: A facilities is created/edited successfully | |
| Fail: Facilities is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to facilities management and click on Create facilities link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit facilities form with following controls:  - “Name” text field  - “Is Active” check box  - “Image” select file button  - “Create” / “Save” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Name” text field blank | Name cannot be blank |

#### DELETE FACILITIES

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.13 | |
| **Use Case Name** | Delete Facilities | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin delete a facilities | |
| **Trigger** | Admin want to delete a facilities | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: Facilities is deleted successfully | |
| Fail: Facilities is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to faculty management and click on Delete button  3. Actor click on “Yes” button | 2. System shows confirm message box with following controls:  - “Yes” button  - “No” button  4. System delete the correspond record in database and redirect to facilities management page |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor click “No” button on confirm message box | No record is deleted and system redirect to facilities management page |

#### CREATE/EDIT STREAM

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.14, UC-2.15 | |
| **Use Case Name** | Create/Edit Stream | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin create or edit a stream | |
| **Trigger** | Admin want to create/edit a stream | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: A Stream is created/edited successfully | |
| Fail: Stream is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to stream management and click on Create Stream link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit facilities form with following controls:  - “Stream Name” text field  - “Create” / “Save” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Stream Name” text field blank | Stream Name cannot be blank |

#### DELETE STREAM

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.13 | |
| **Use Case Name** | Delete Stream | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin delete a stream | |
| **Trigger** | Admin want to delete a stream | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: Stream is deleted successfully | |
| Fail: Stream is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to stream management and click on Delete button  3. Actor click on “Yes” button | 2. System shows confirm message box with following controls:  - “Yes” button  - “No” button  4. System delete the correspond record in database and redirect to stream management page |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor click “No” button on confirm message box | No record is deleted and system redirect to stream management page |

#### CREATE/EDIT FIELD

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.14, UC-2.15 | |
| **Use Case Name** | Create/Edit Field | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin create or edit a field | |
| **Trigger** | Admin want to create/edit a field | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: A field is created/edited successfully | |
| Fail: Field is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to field management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit facilities form with following controls:  - “Field Name” text field  - “Stream” select box  - “Create” / “Save” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Field Name” text field blank | Field Name cannot be blank |
| Actor left “Stream” select box | Stream cannot be blank |

#### DELETE FIELD

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.13 | |
| **Use Case Name** | Delete Field | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin delete a field | |
| **Trigger** | Admin want to delete a field | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: Field is deleted successfully | |
| Fail: Field is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to field management and click on Delete button  3. Actor click on “Yes” button | 2. System shows confirm message box with following controls:  - “Yes” button  - “No” button  4. System delete the correspond record in database and redirect to field management page |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor click “No” button on confirm message box | No record is deleted and system redirect to field management page |

#### CREATE/EDIT SPECIALIZED SUBJECT

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.14, UC-2.15 | |
| **Use Case Name** | Create/Edit Specialized Subject | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin create or edit a specialized subject | |
| **Trigger** | Admin want to create/edit a specialized subject | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: A specialized subject is created/edited successfully | |
| Fail: Specialized subject is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to specialized subject management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit facilities form with following controls:  - “Subject Name” text field  - “Stream” select box  - “Field” select box  - “Create” / “Save” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Subject Code” text field blank | Subject Code cannot be blank |
| Actor left “Subject Name” text field blank | Subject Code cannot be blank |
| Actor left “Stream” select box | Stream cannot be blank |
| Actor left “Field” select box | Field cannot be blank |
| Actor input an existed subject code | Subject Code is existed. Please choose another one |

#### DELETE SPECIALIZED SUBJECT

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.13 | |
| **Use Case Name** | Delete specialized subject | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin delete a specialized subject | |
| **Trigger** | Admin want to delete a specialized subject | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: Specialized subject is deleted successfully | |
| Fail: Specialized subject is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to specialized subject management and click on Delete button  3. Actor click on “Yes” button | 2. System shows confirm message box with following controls:  - “Yes” button  - “No” button  4. System delete the correspond record in database and redirect to specialized subject management page |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor click “No” button on confirm message box | No record is deleted and system redirect to specialized subject management page |

#### CREATE/EDIT OPTIONAL SUBJECT

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.14, UC-2.15 | |
| **Use Case Name** | Create/Edit Optional Subject | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin create or edit an optional subject | |
| **Trigger** | Admin want to create/edit an optional subject | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: A optional subject is created/edited successfully | |
| Fail: Optional subject is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to optional subject management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit facilities form with following controls:  - “Subject Code” text field  - “Subject Name” text field  - “Create” / “Save” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Subject Code” text field blank | Subject Code cannot be blank |
| Actor left “Subject Name” text field blank | Subject Code cannot be blank |
| Actor input an existed subject code | Subject Code is existed. Please choose another one |

#### DELETE OPTIONAL SUBJECT

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.13 | |
| **Use Case Name** | Delete optional subject | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin delete an optional subject | |
| **Trigger** | Admin want to delete an optional subject | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: Optional subject is deleted successfully | |
| Fail: Optional subject is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to optional subject management and click on Delete button  3. Actor click on “Yes” button | 2. System shows confirm message box with following controls:  - “Yes” button  - “No” button  4. System delete the correspond record in database and redirect to optional subject management page |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor click “No” button on confirm message box | No record is deleted and system redirect to optional management page |

#### CREATE/EDIT ACCOUNT

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.14, UC-2.15 | |
| **Use Case Name** | Create/Edit Account | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin create or edit an account subject | |
| **Trigger** | Admin want to create/edit an account | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: A account is created/edited successfully | |
| Fail: Account is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to account management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit facilities form with following controls:  - “Full Name” text field  - “Email” text field  - “Username” text field  - “Password” text field  - “Role” select box  - “Is Active” check box  - “Create” / “Save” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Full Name” text field blank | Full Name cannot be blank |
| Actor left “Email” text field | Email cannot be blank |
| Actor left “Username” text field | Username cannot be blank |
| Actor left “Password” text field blank | Password cannot be blank |
| Actor left “Role” select box | Role cannot be blank |
| Actor input invalid email address | Email is invalid |
| Actor input an existed username | Username is existed. Please choose another one |

#### DELETE ACCOINT

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.13 | |
| **Use Case Name** | Delete account | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin delete an account | |
| **Trigger** | Admin want to delete an account | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: Account is deleted successfully | |
| Fail: Account is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to account management and click on Delete button  3. Actor click on “Yes” button | 2. System shows confirm message box with following controls:  - “Yes” button  - “No” button  4. System delete the correspond record in database and redirect to account management page |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor click “No” button on confirm message box | No record is deleted and system redirect to account management page |

#### LOGIN

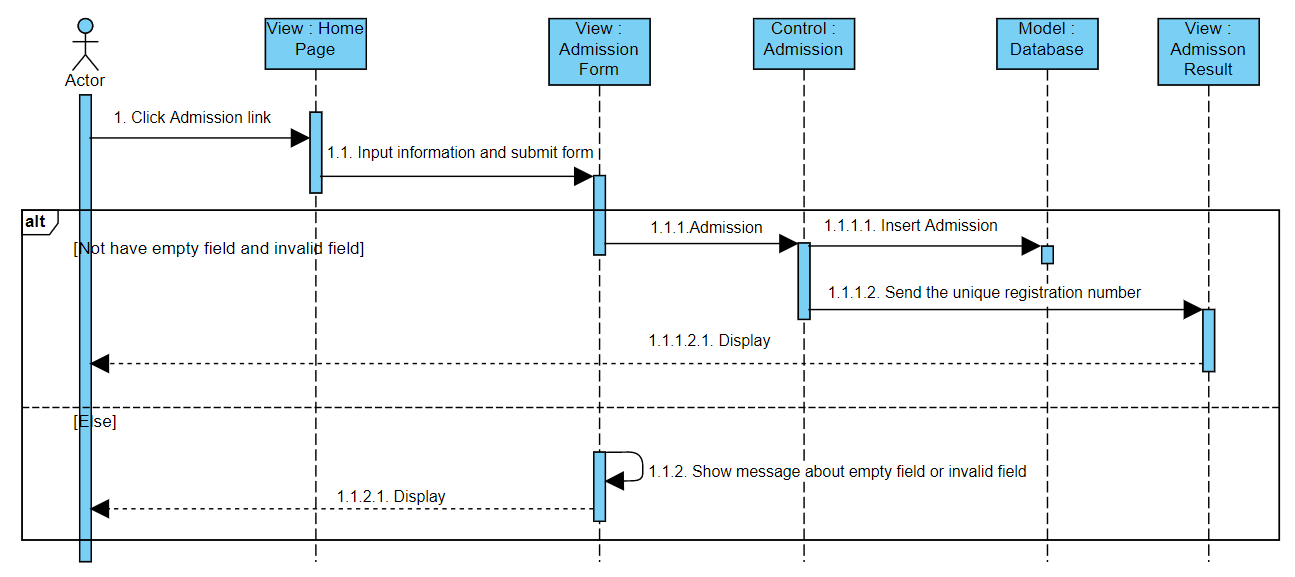
|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.14, UC-2.15 | |
| **Use Case Name** | Login | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin login to admin system | |
| **Trigger** | Admin want to login to admin system | |
| **Pre-conditions** | The admin’s device is already connected to the internet | |
| **Post-conditions** | Success: Admin login successfully | |
| Fail: Admin login fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to admin system  3. Actor fill the information and click on “Login” button | 2. System shows the login form with following controls:  - “Username” text field  - “Password” text field  - “Login” button  4. System validates the information  5. System redirect to admin system |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Username” text field blank | Username cannot be blank |
| Actor left “Password” text field blank | Password cannot be blank |
| Actor input an incorrect login details | Username or Password is incorrect |
| Account is not active | Account is not active |

#### LOG OUT

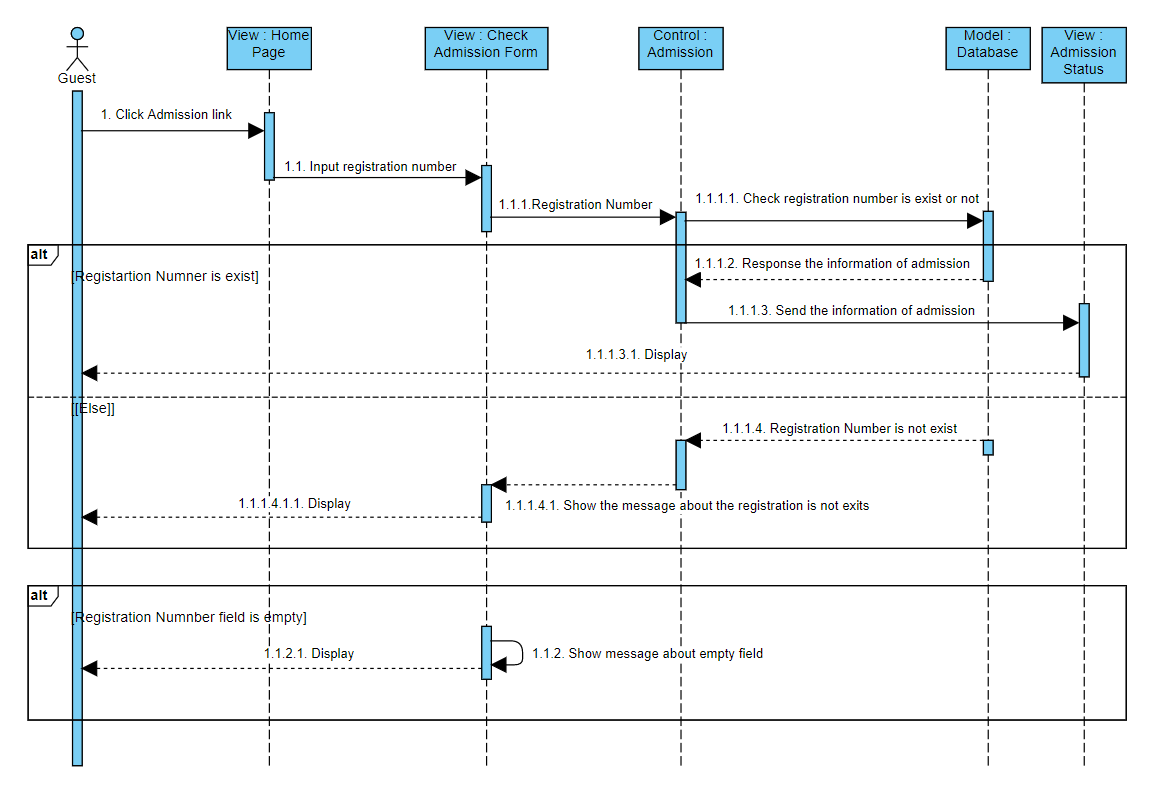
|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.13 | |
| **Use Case Name** | Log out | |
| **Author** | All team | |
| **Actors** | Admin | |
| **Description** | Admin log out of admin system | |
| **Trigger** | Admin want to log out of admin system | |
| **Pre-conditions** | The admin’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: Log out successfully | |
| Fail: Log out fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to Log out link | 2. System redirect to login page of admin system. |

### SEQUENCE DIAGRAMS

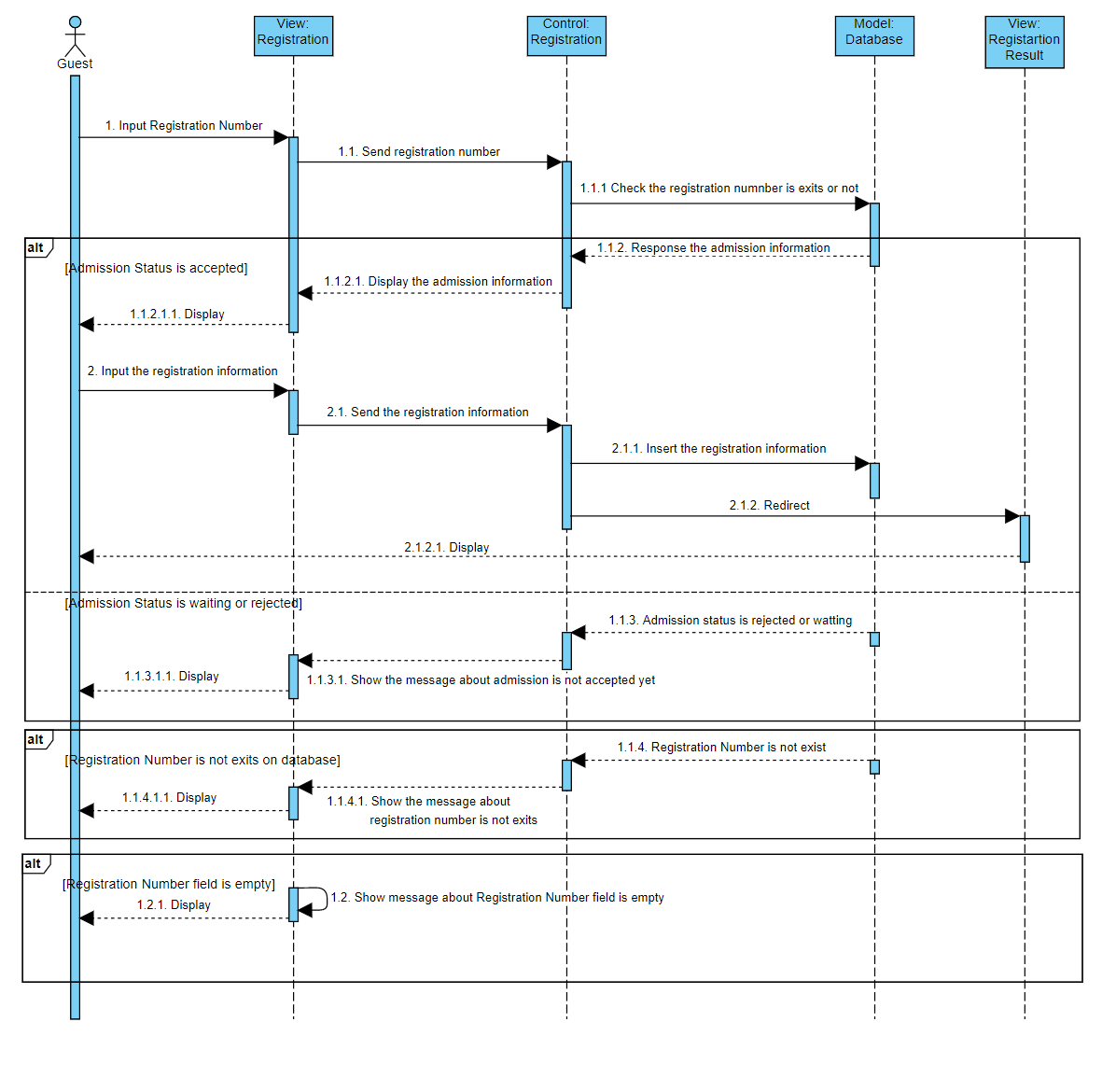
#### ADMISSION



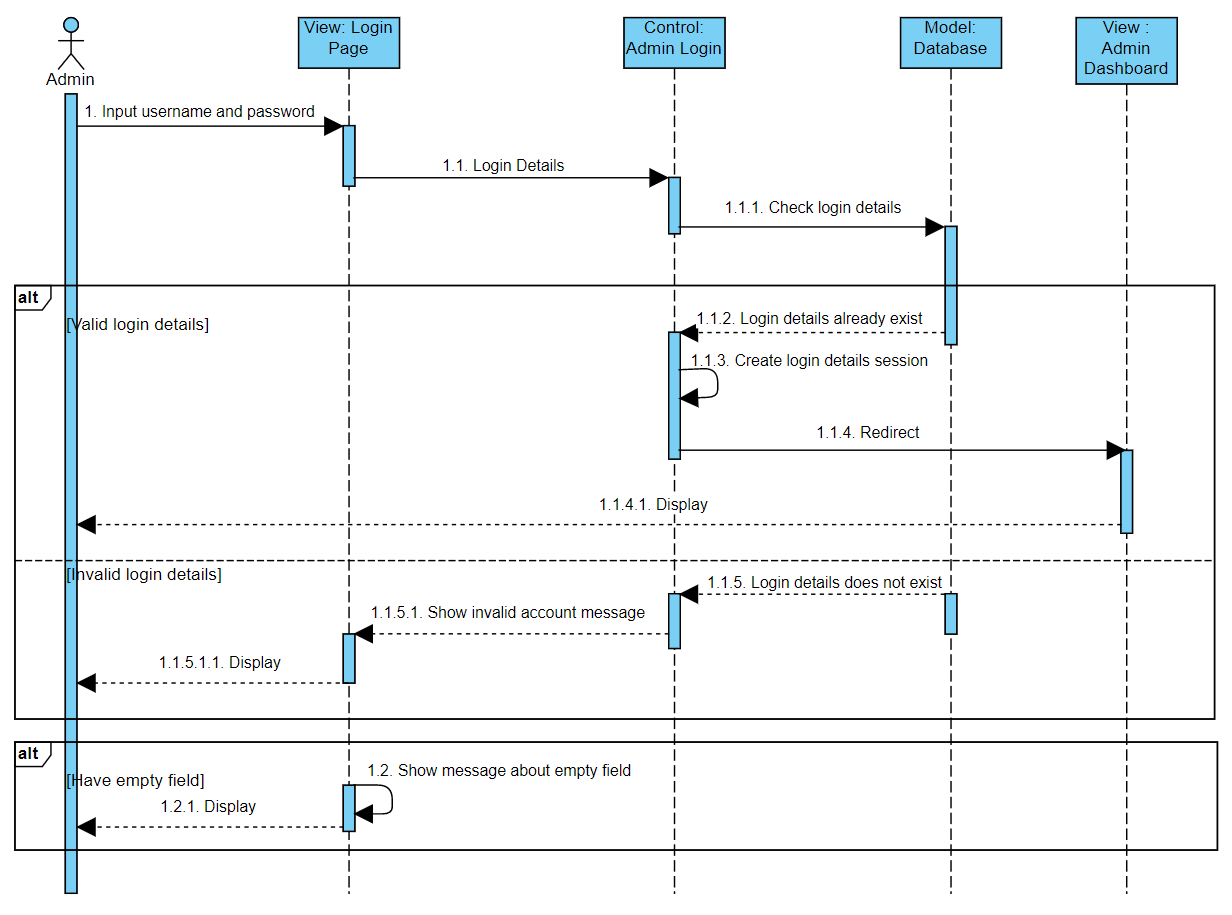
#### CHECK ADMISSION STATUS



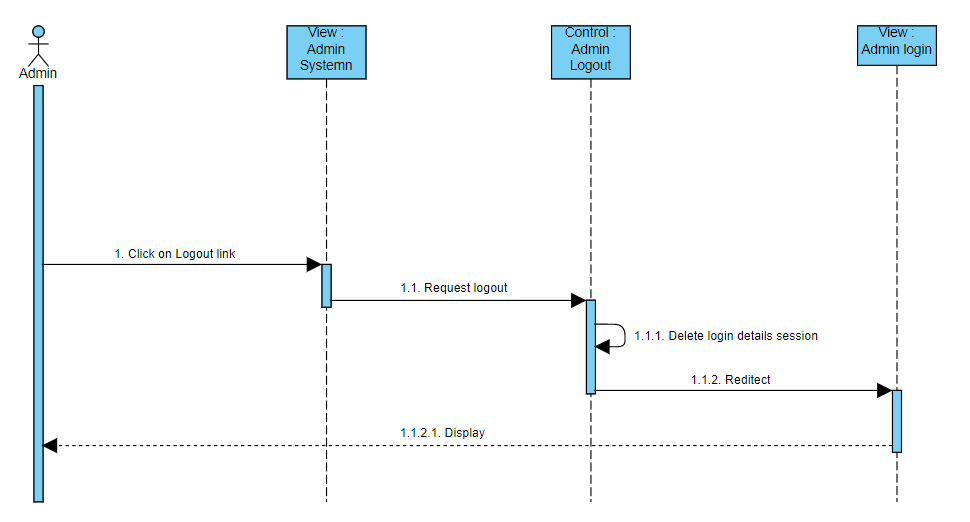
#### STUDENT REGISTRATION



#### ADMIN LOGIN

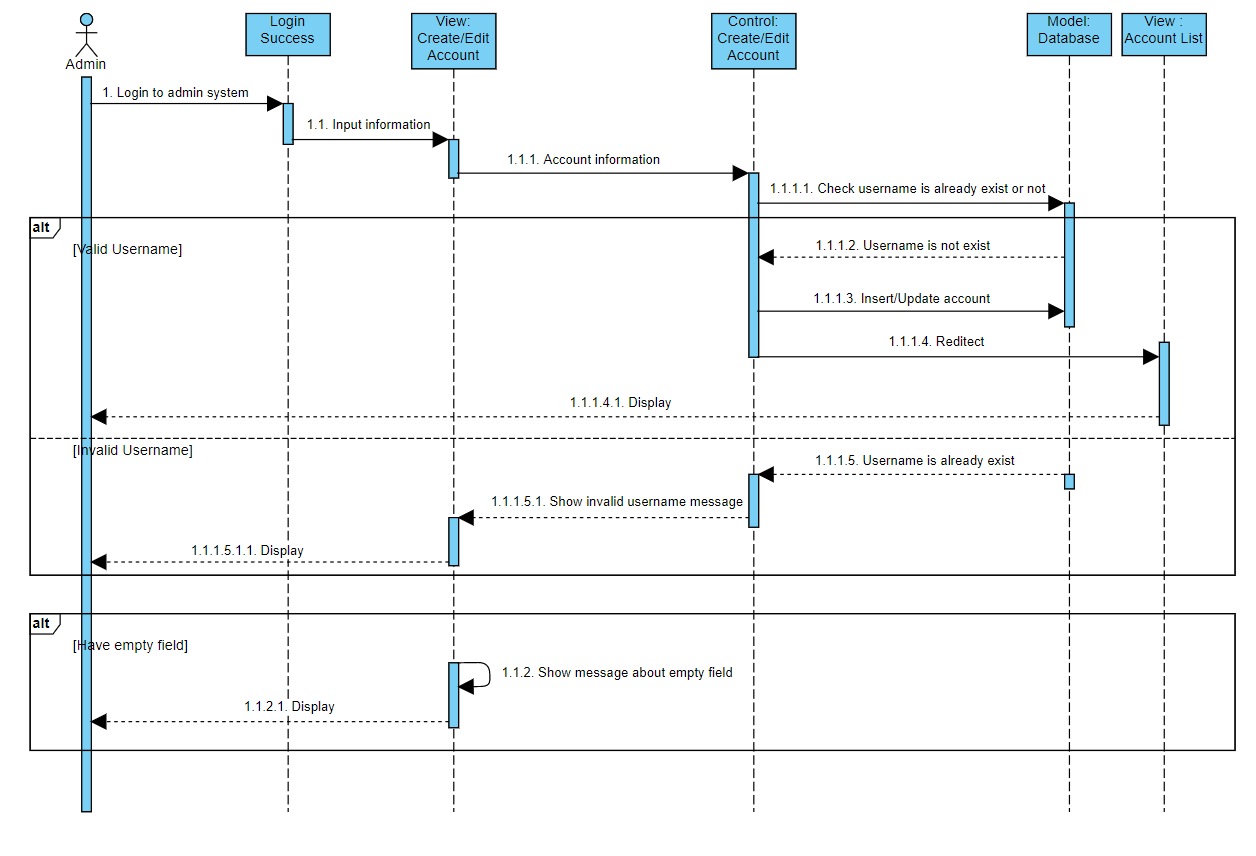


#### ADMIN LOGOUT



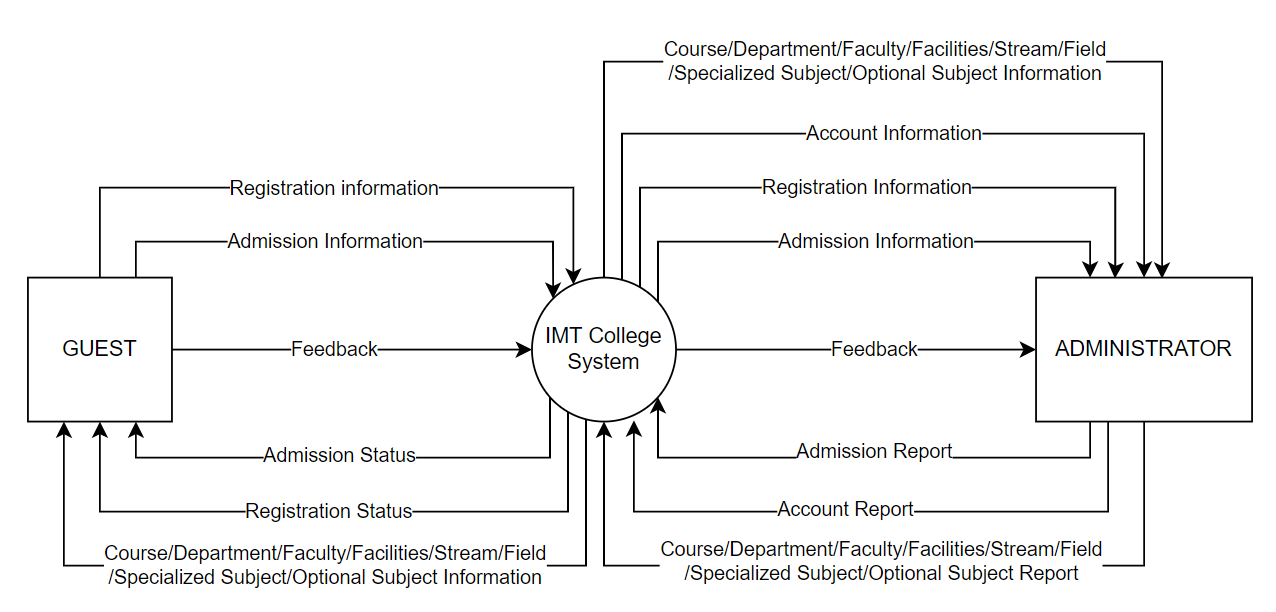
#### ADMIN SYSTEM MANAGEMENT

#### CREATE/EDIT ADMIN ACCOUNT

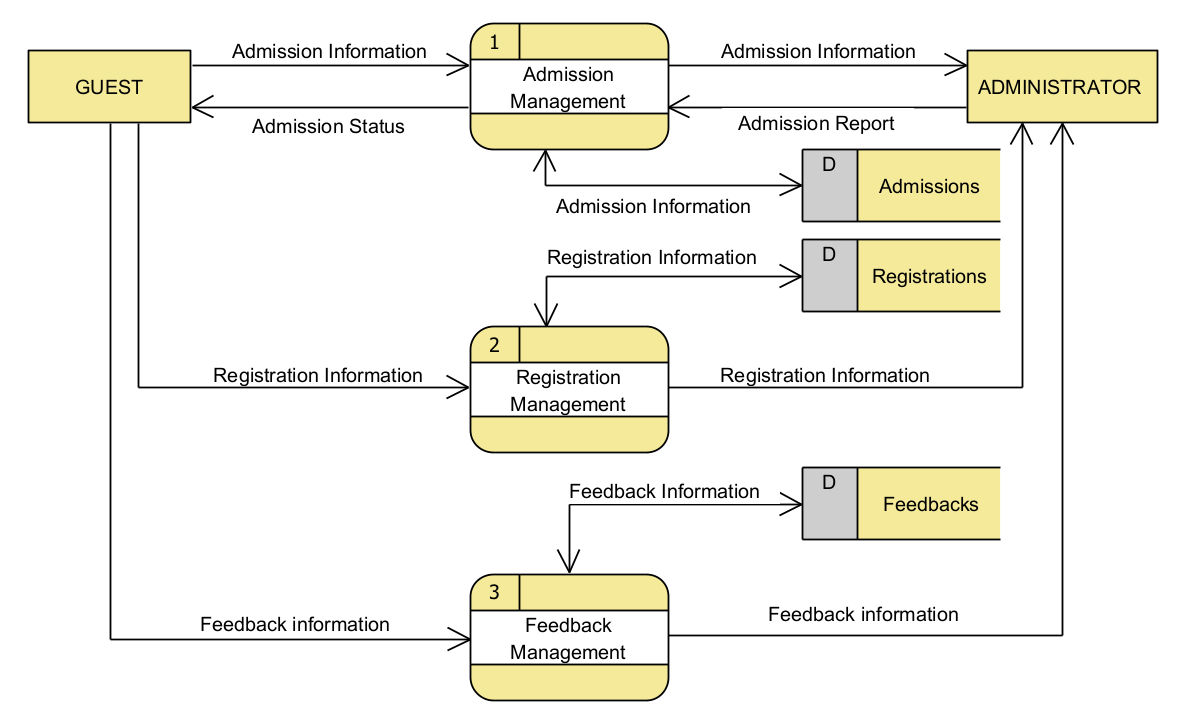


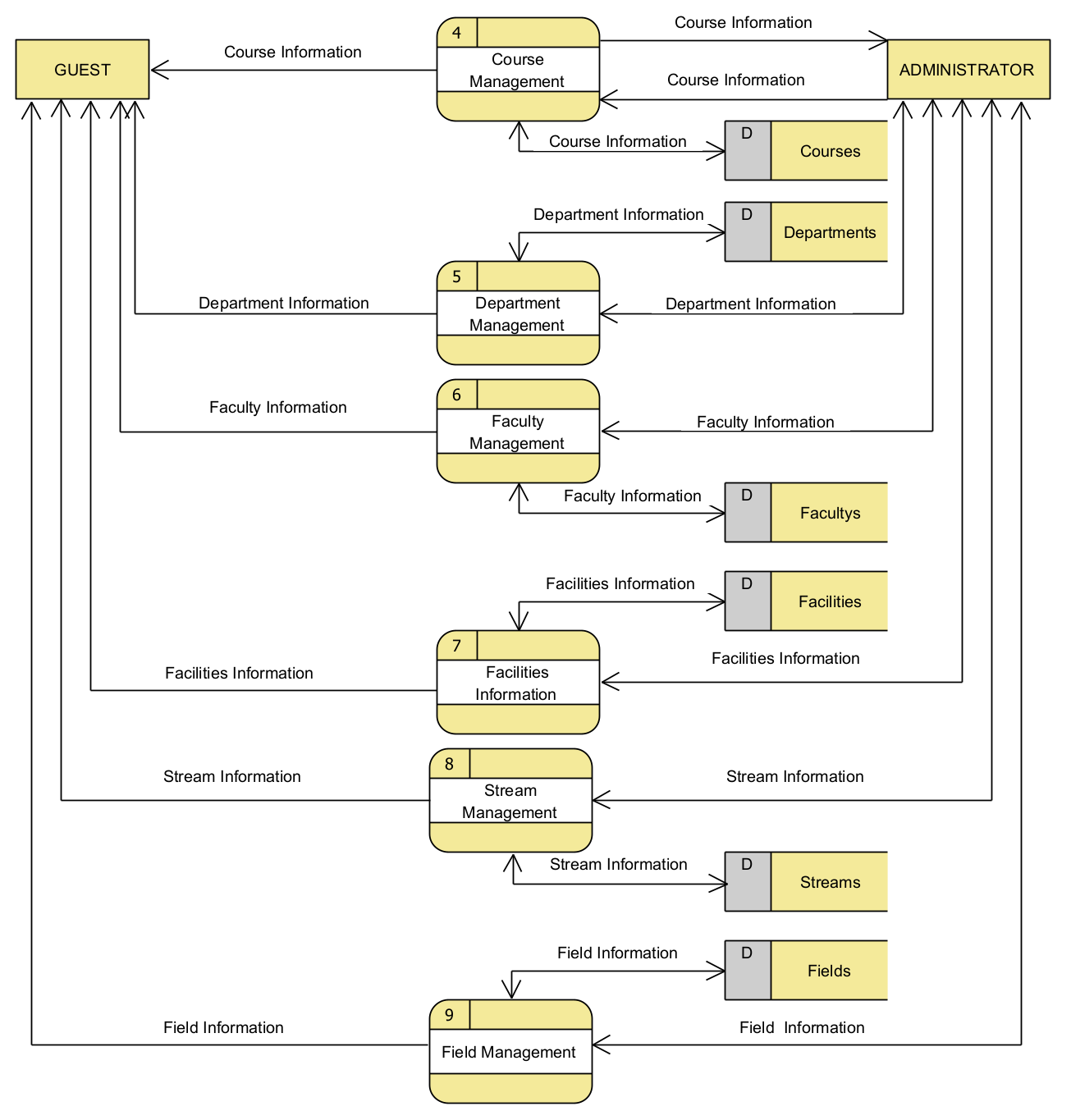
### DATA FLOW DIAGRAM

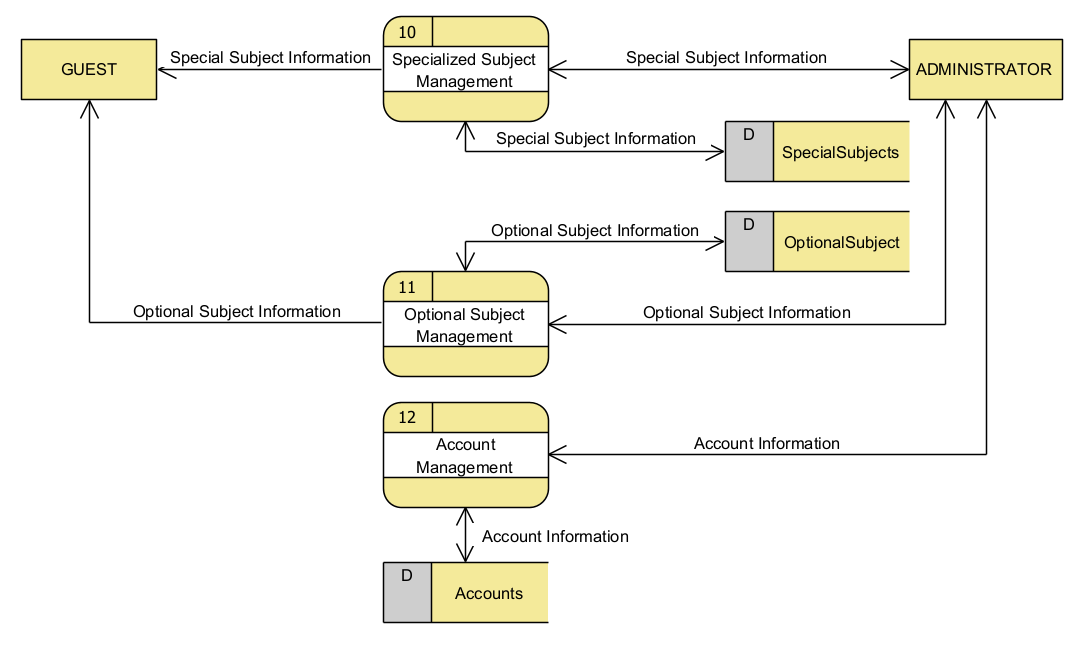
#### CONTEXT DIAGRAM



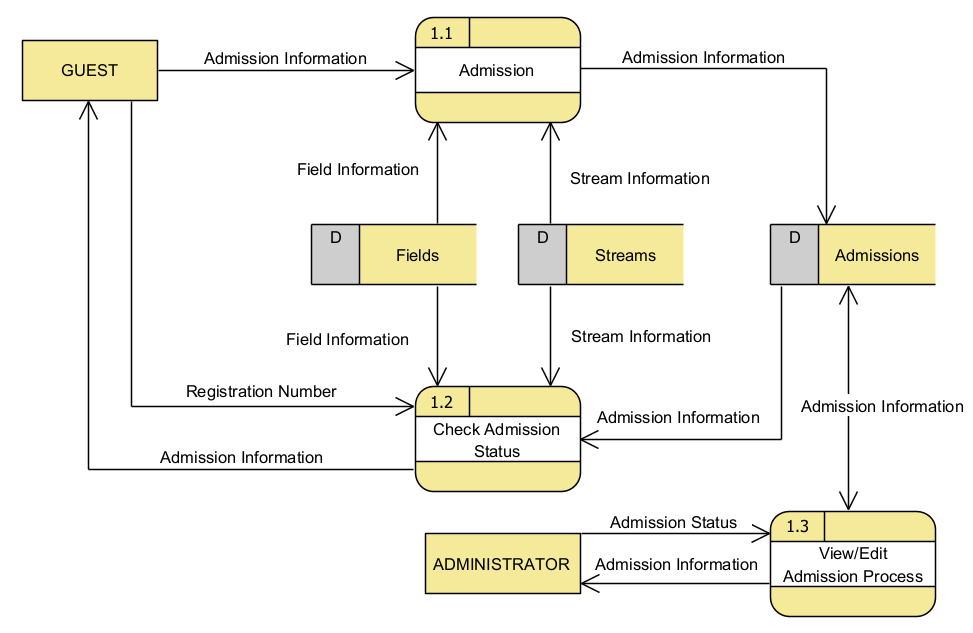
#### LEVEL 0

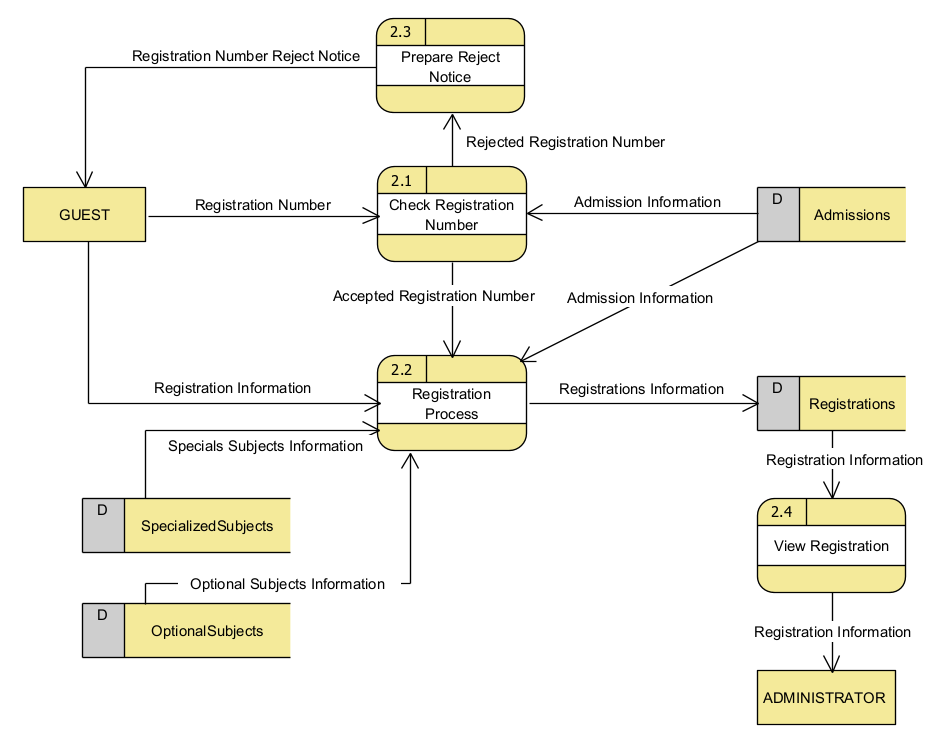


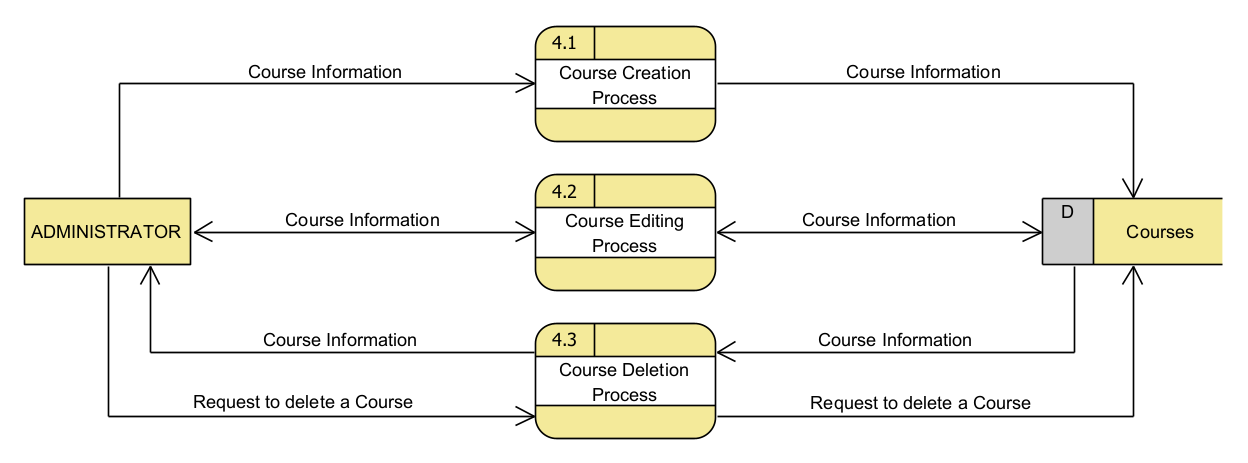


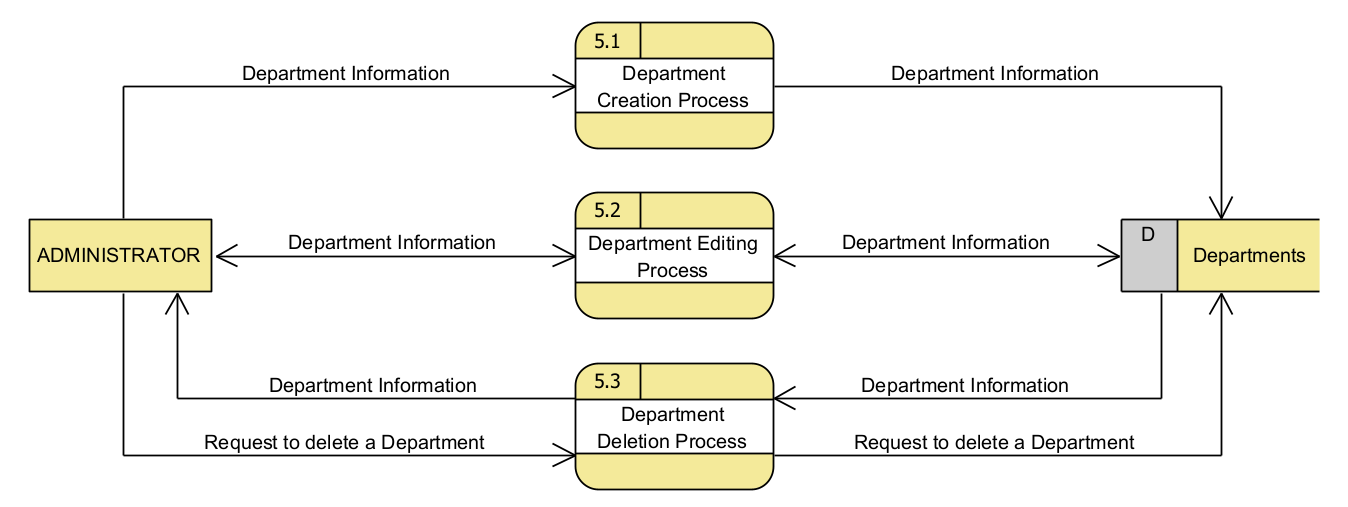


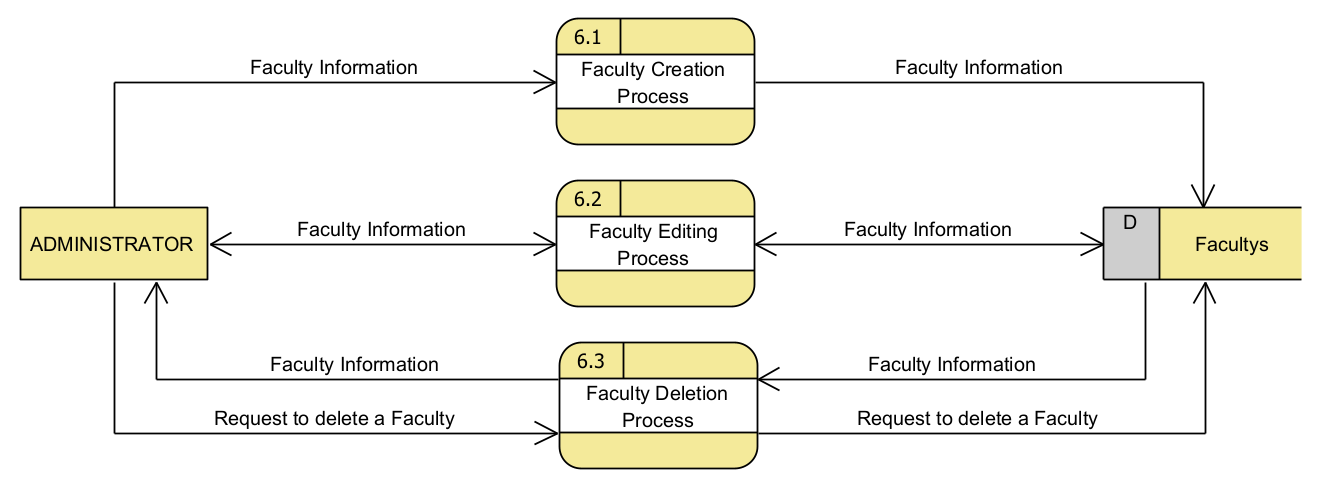
#### LEVEL 1

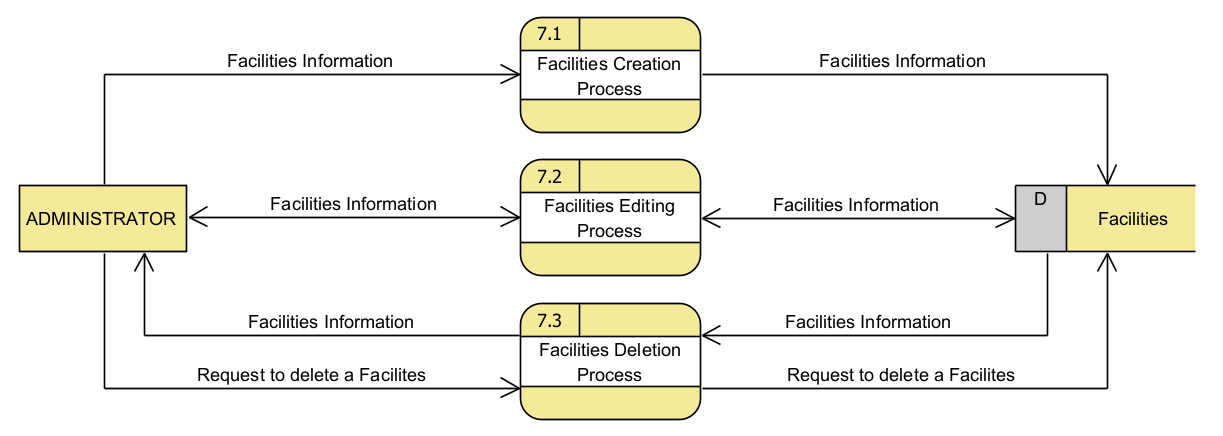


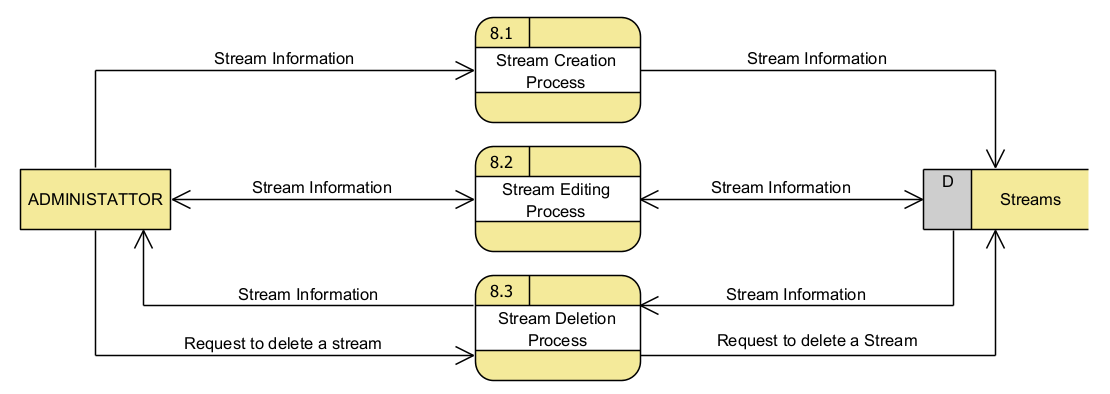


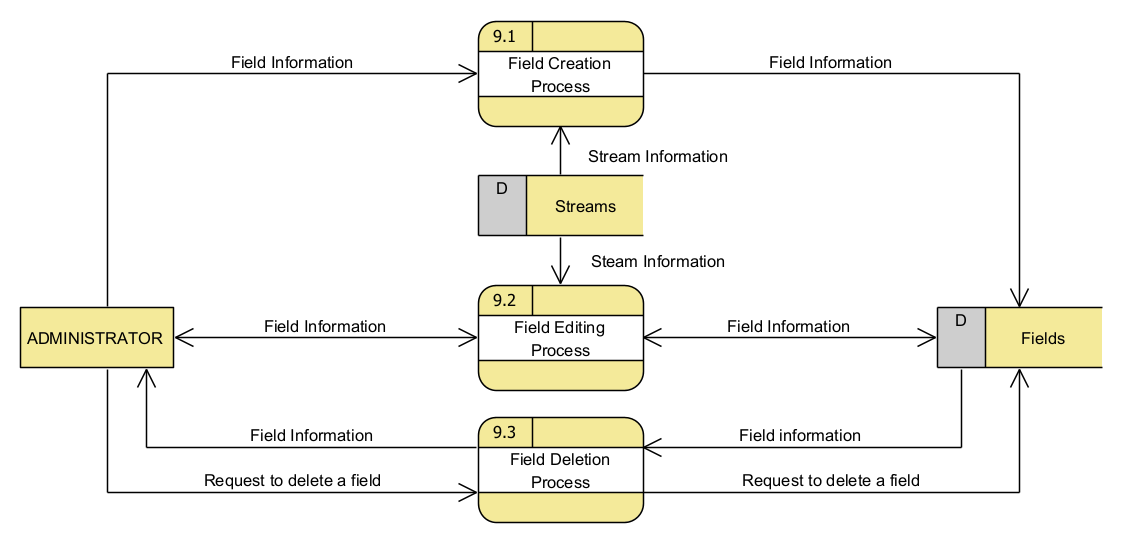


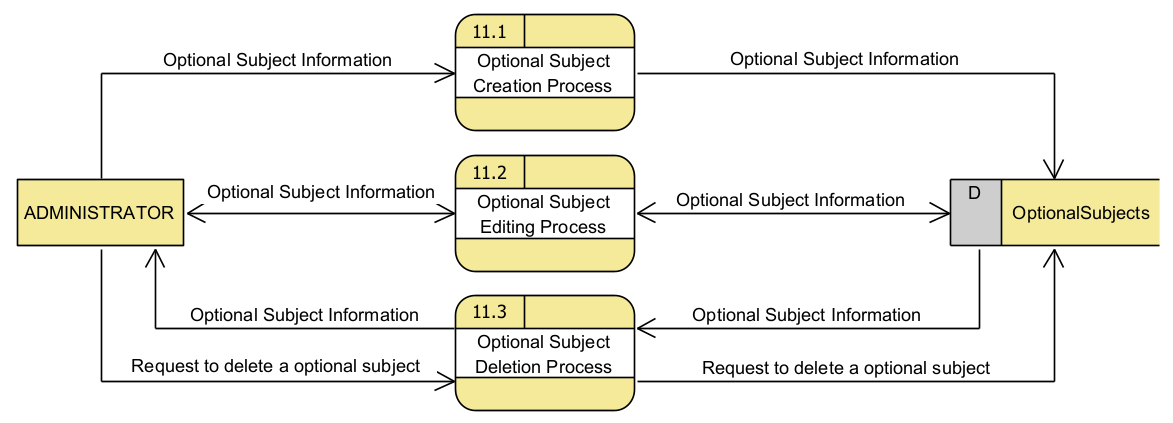


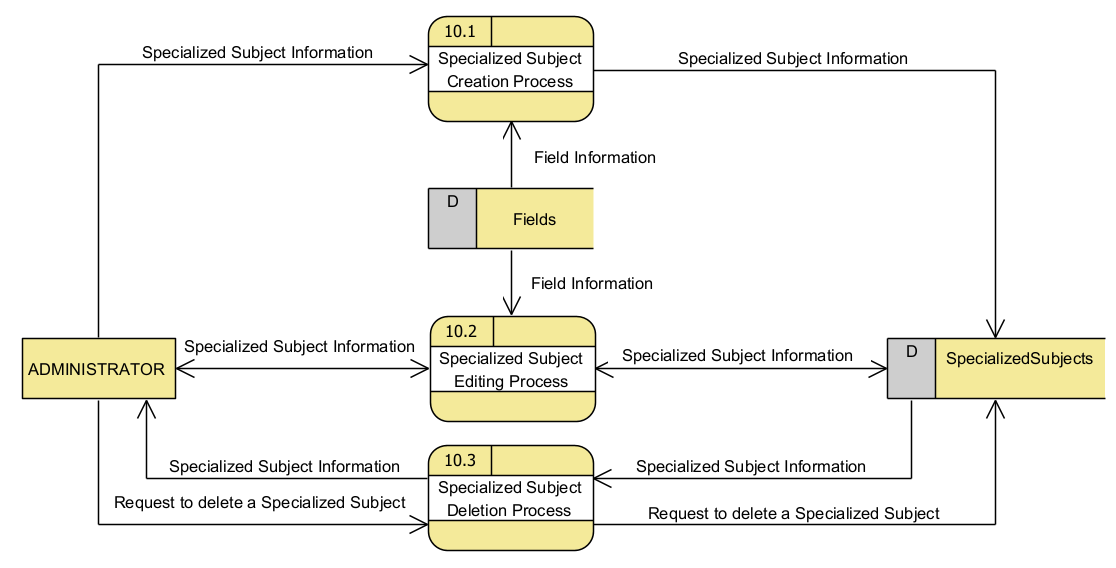


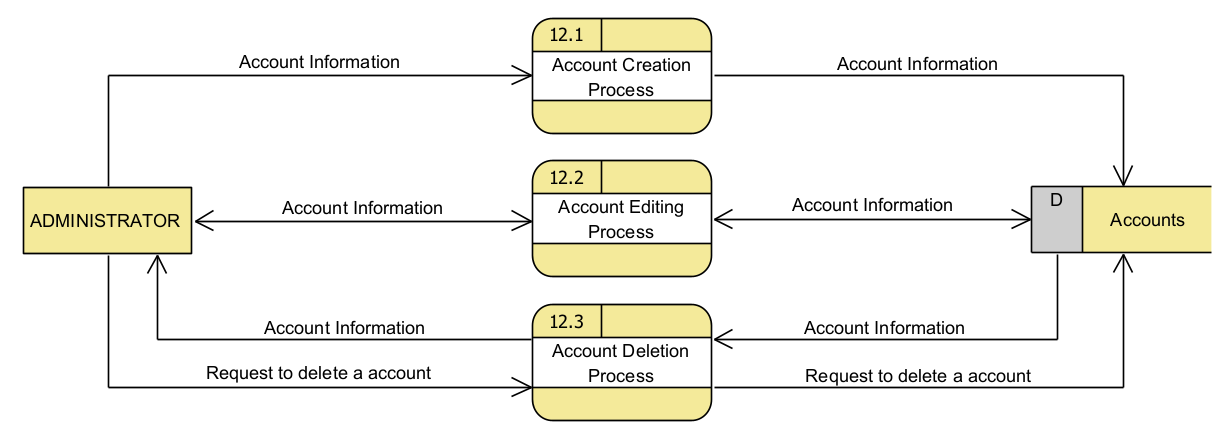


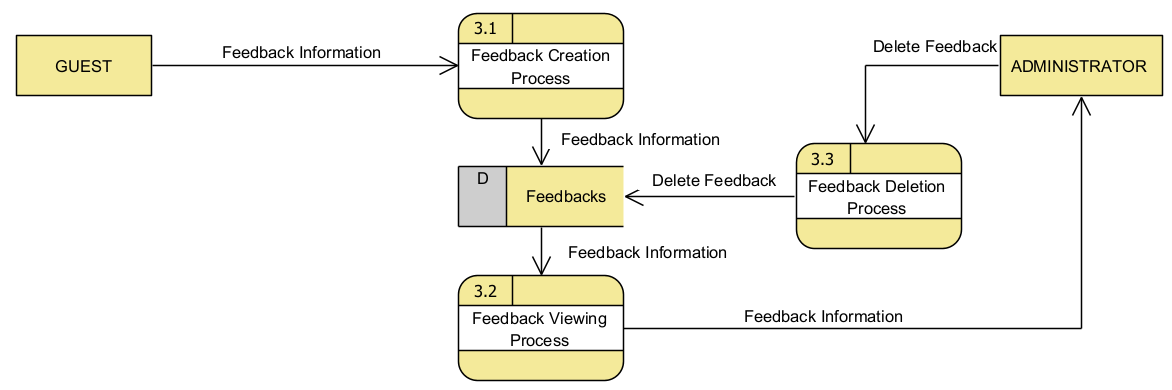




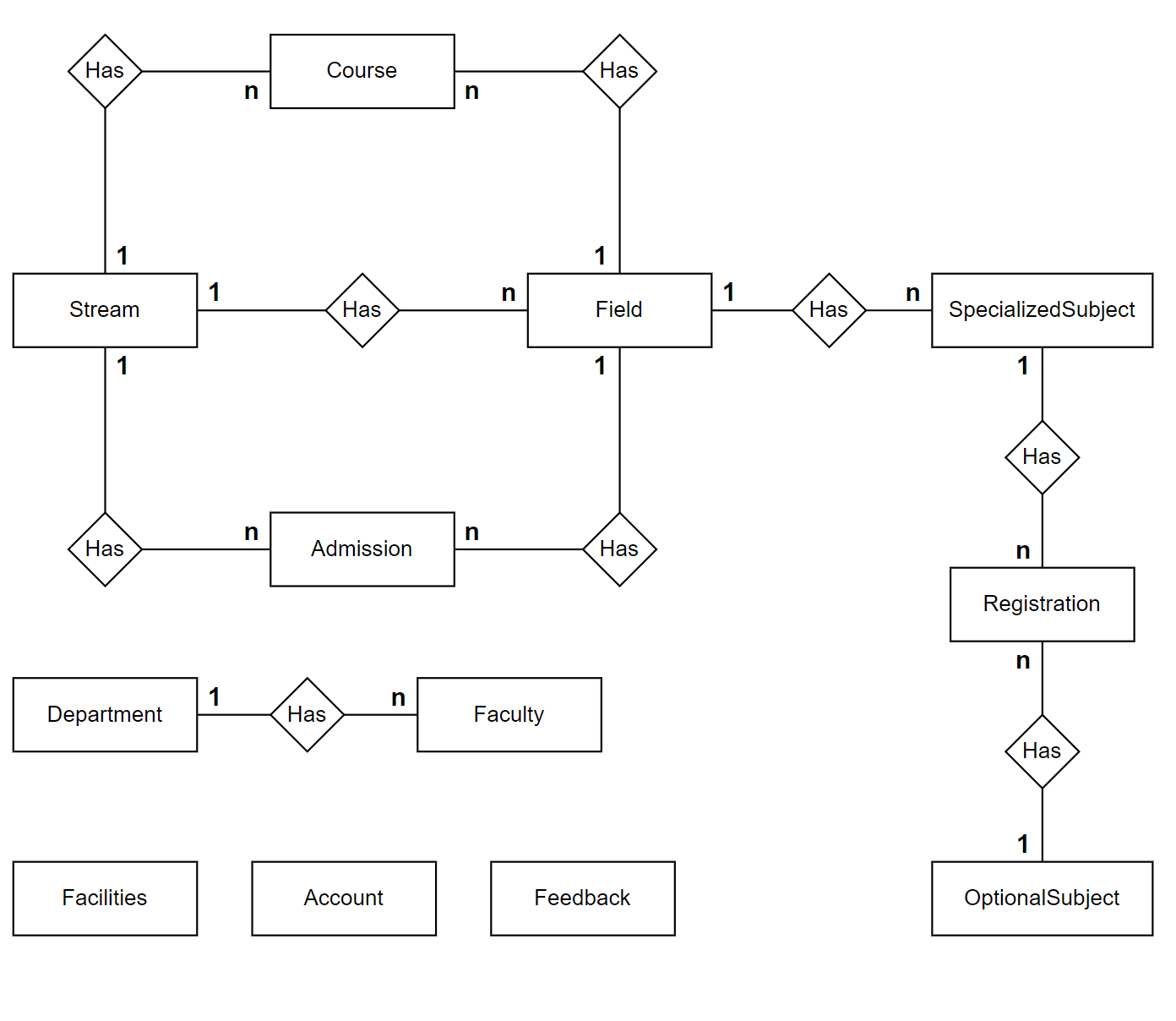






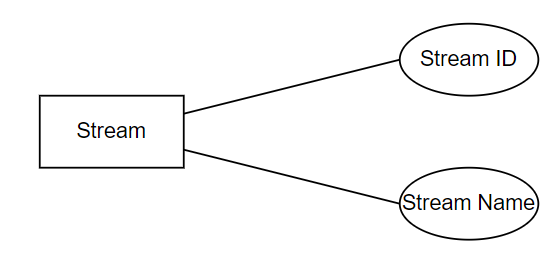


### LOGICAL ENTITY RELATIONSHIP DIAGRAM

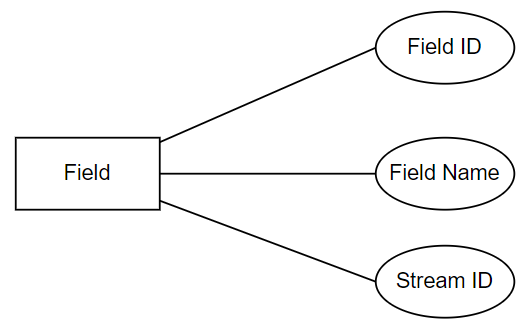


### ENTITY PROPERTIES

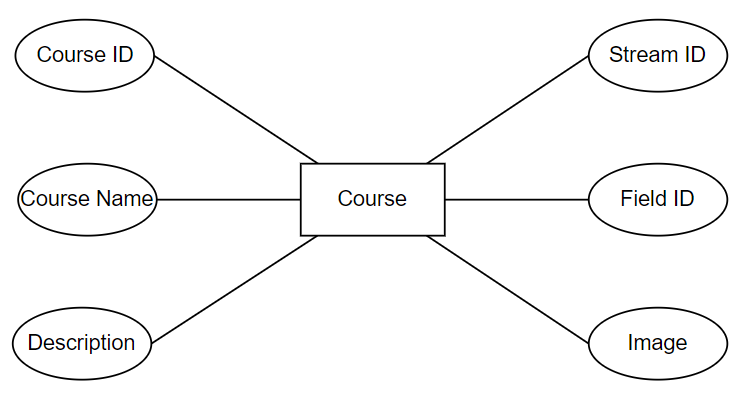
#### STREAM



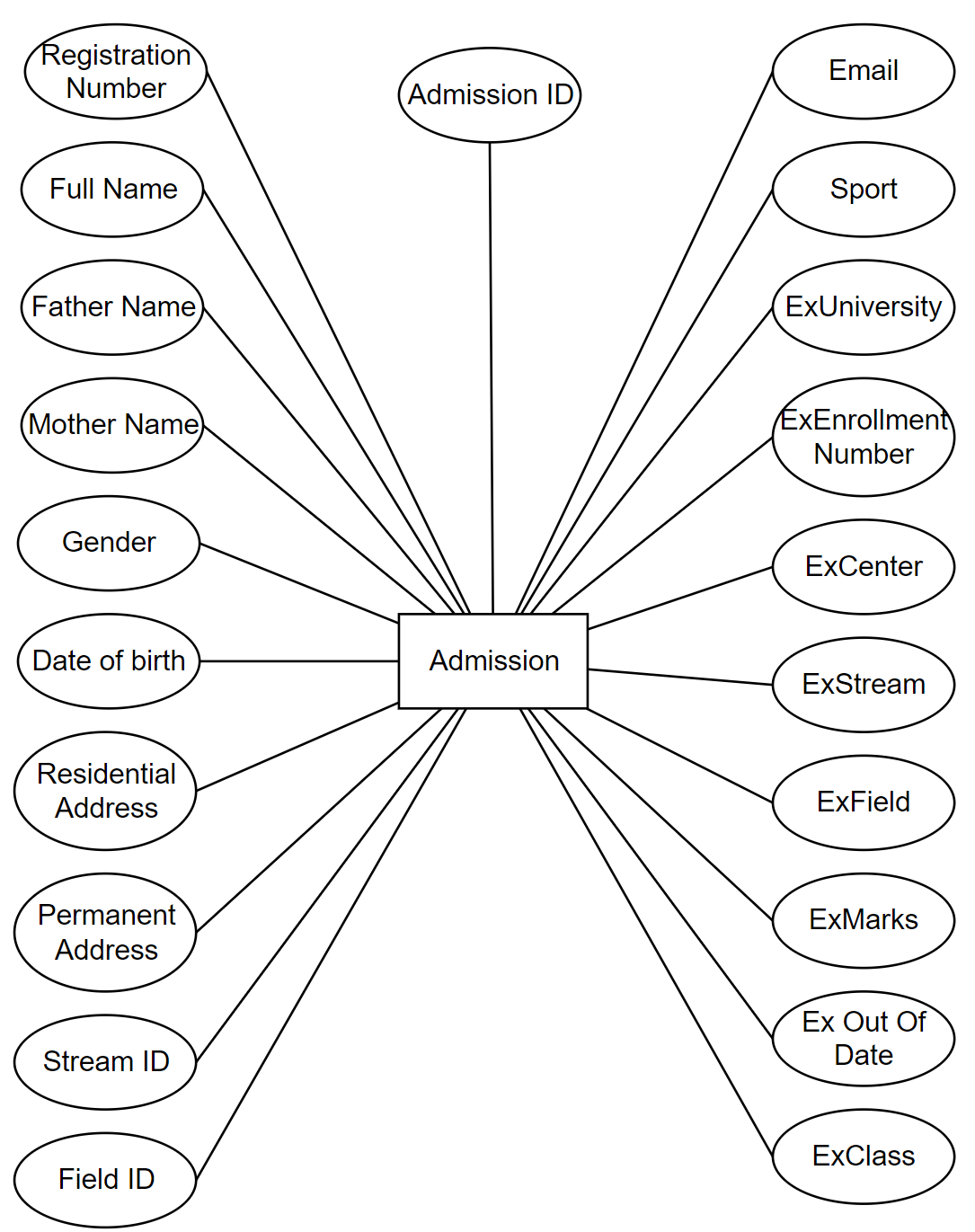
#### FIELD



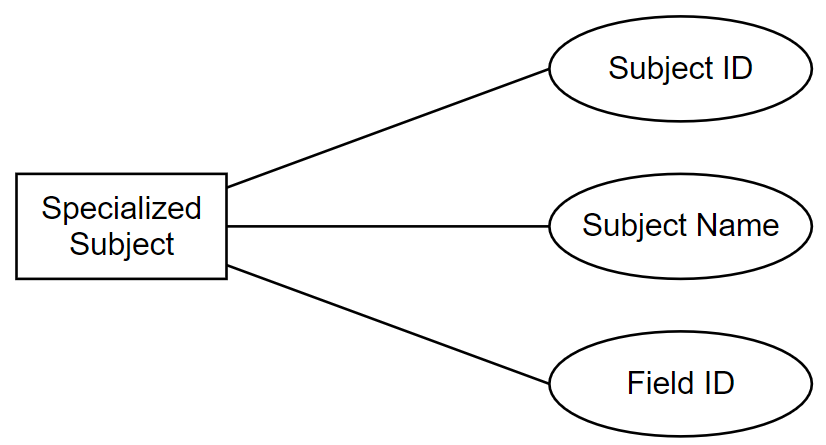
#### COURSE



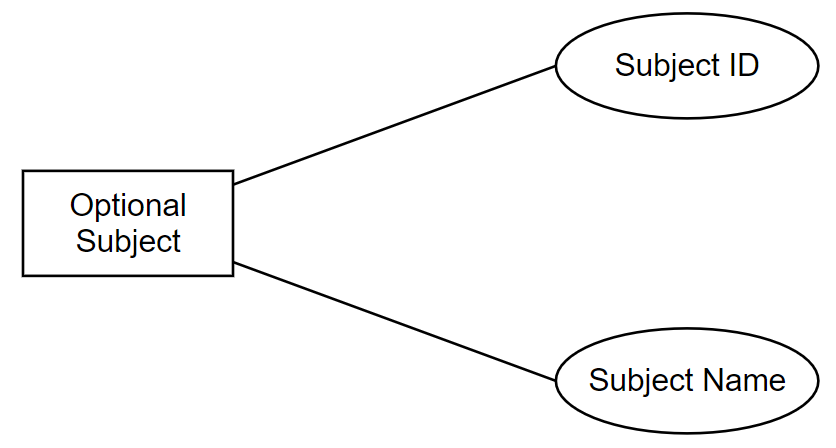
#### ADMISSION



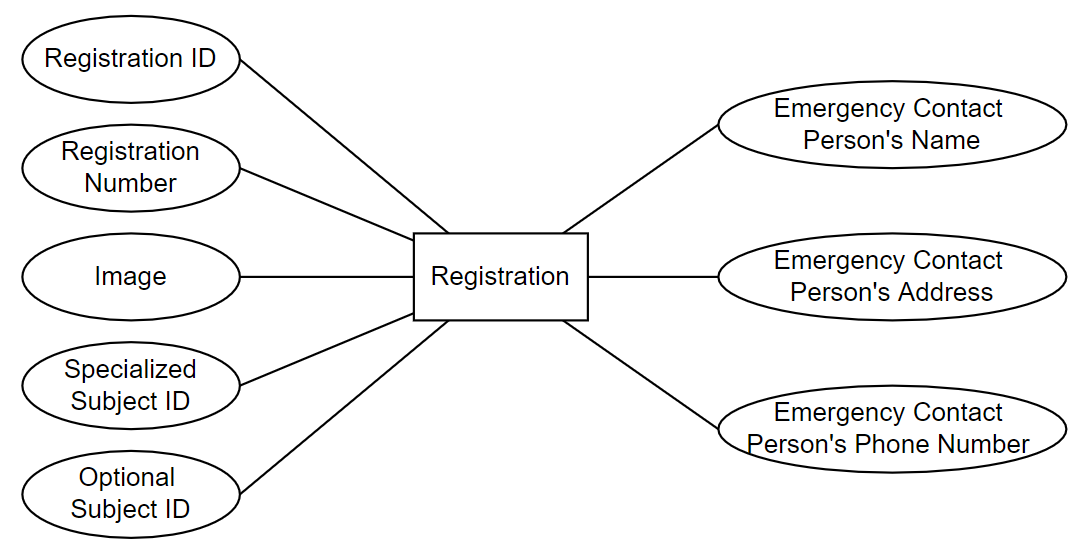
#### SPECIALIZED SUBJECT



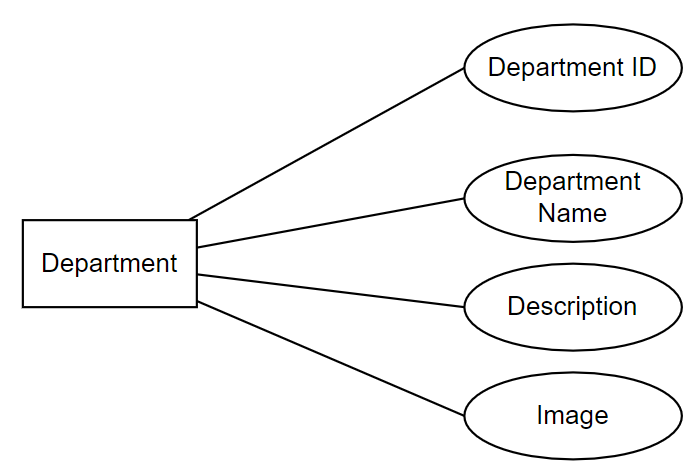
#### OPTIONAL SUBJECT



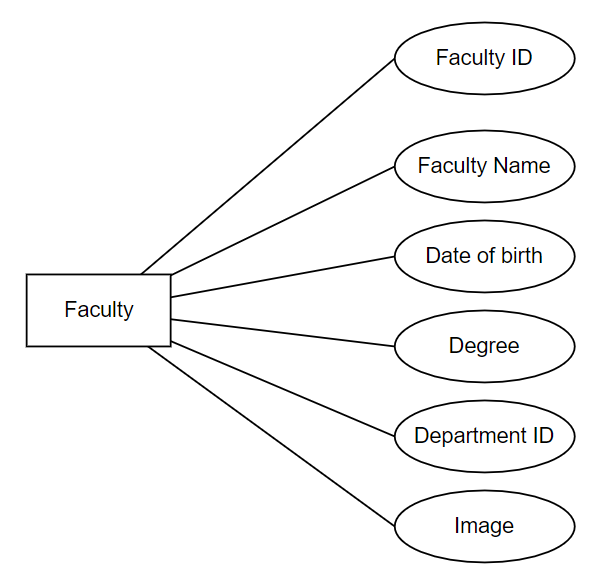
#### REGISTRATION



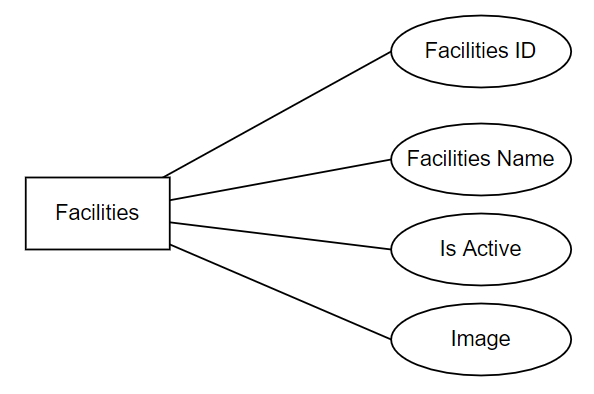
#### DEPARTMENT



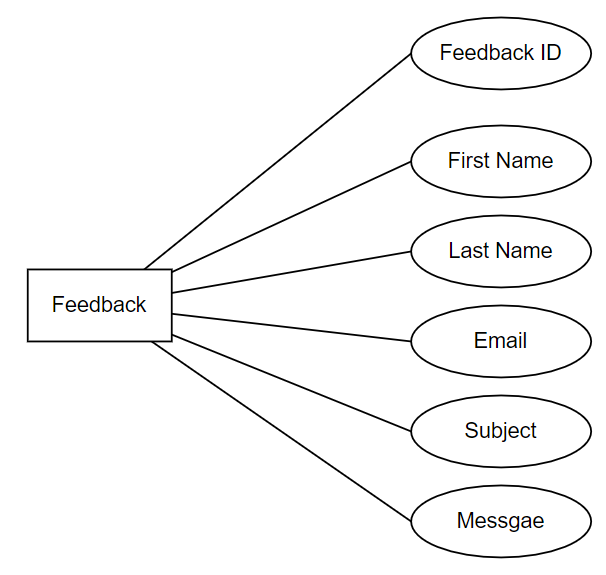
#### FACULTY



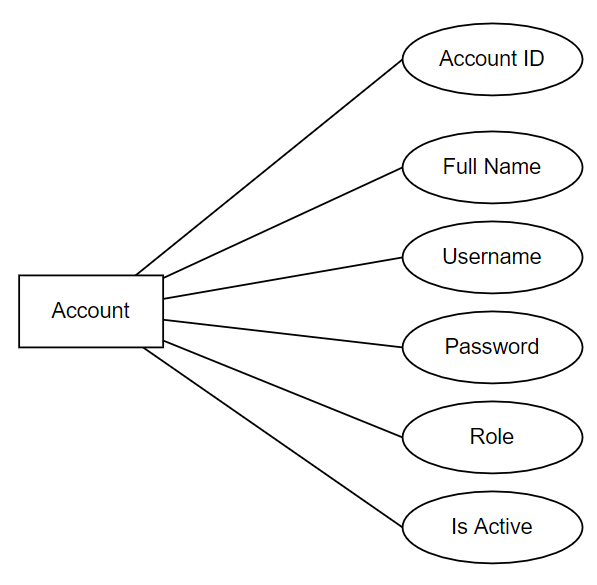
#### FACILITIES



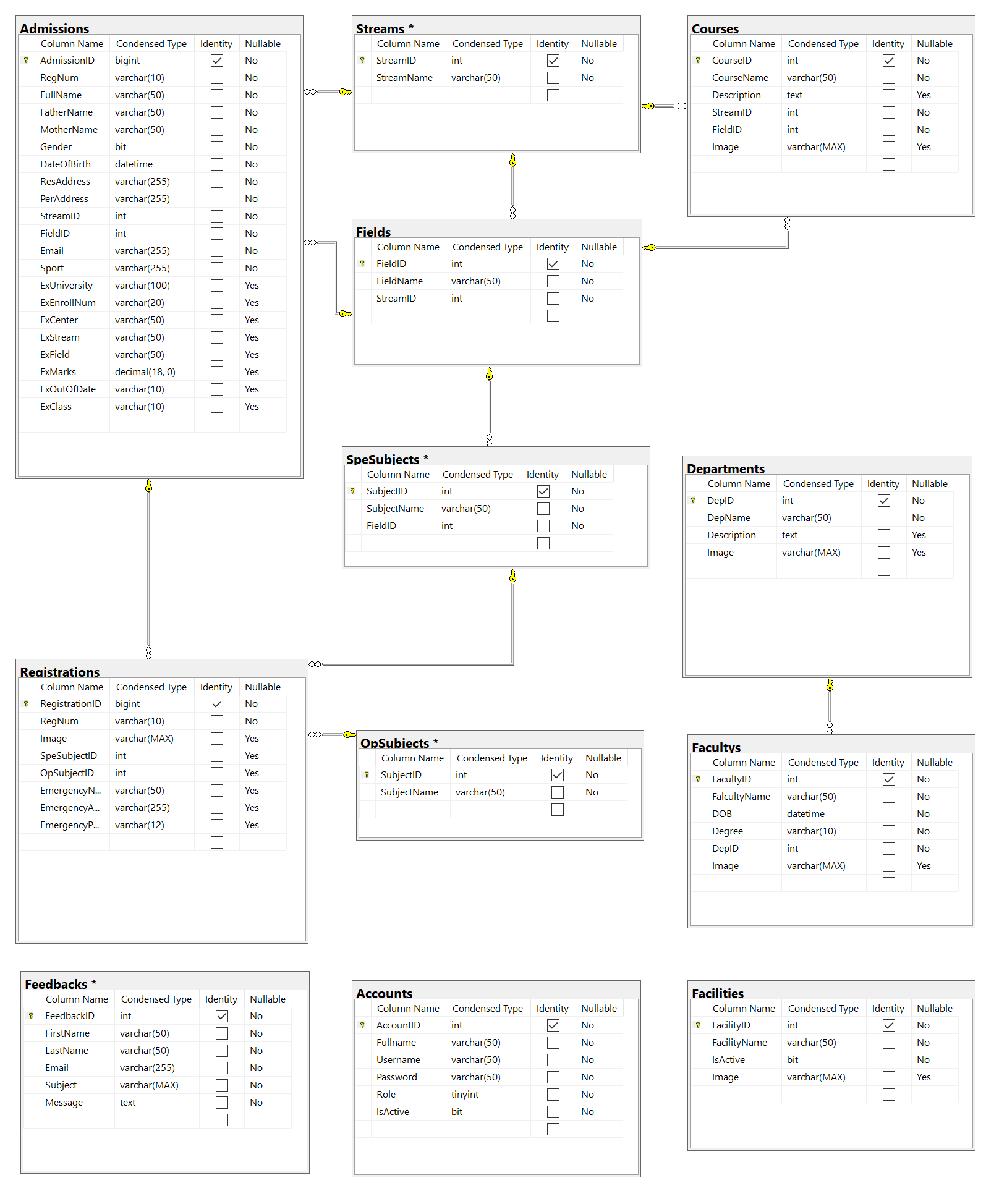
#### FEEDBACK



#### ACCOUNT



### DATABASE DIAGRAM



### TABLE DESIGN

#### STREAM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Streams** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | StreamID | int  identity(1,1) | PRIMARY KEY | Store the ID of streams |
| 2 | StreamName | varchar(50) | NOT NULL | Store the Stream Name |

#### FIELD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | FieldID | int  identity(1,1) | PRIMARY KEY | Store the ID of field |
| 2 | FieldName | varchar(50) | NOT NULL | Store the Field Name |
| 3 | StreamID | int | NOT NULL  FOREIGN KEY  REFERENCES  COLUMN “StreamID” ON “Streams” TABLE | Store the Stream ID |

#### COURSE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | CourseID | int  identity(1,1) | PRIMARY KEY | Store the ID of course |
| 2 | CourseName | varchar(50) | NOT NULL | Store the Course Name |
| 3 | Description | Text |  | Store the description of course |
| 4 | StreamID | int | NOT NULL  FOREIGN KEY  REFERENCES  COLUMN “StreamID” ON “Streams” TABLE | Store the Stream ID |
| 5 | FieldID | int | NOT NULL  FOREIGN KEY  REFERENCES  COLUMN “FieldID” ON “Fields” TABLE | Store the Field ID |
| 6 | Image | varchar(max) |  | Store the URL of image |

#### ADMISSION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Admissions** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | AdmissonID | bigint  identity(1,1) | PRIMARY KEY | Store the ID of admission |
| 2 | RegNum | varchar(10) | NOT NULL | Store the Registration Number |
| 3 | Fullname | varchar(50) | NOT NULL | Store the Full Name of student |
| 4 | FatherName | varchar(50) | NOT NULL | Store the name of student’s father |
| 5 | MotherName | varchar(50) | NOT NULL | Store the name of student’s mother |
| 6 | Gender | bit | NOT NULL | Store the sexes of student |
| 7 | DateOfBirth | datetime | NOT NULL | Store the student’s date of birth |
| 8 | ResAddress | varchar(255) | NOT NULL | Store the student’s residential address |
| 9 | PerAddress | varchar(255) | NOT NULL | Store the student’s permanent address |
| 10 | StreamID | int | NOT NULL  FOREIGN KEY  REFERENCES  COLUMN “StreamID” ON “Streams” TABLE | Store the Stream ID |
| 11 | FieldID | int | NOT NULL  FOREIGN KEY  REFERENCES  COLUMN “FieldID” ON “Fields” TABLE | Store the Field ID |
| 12 | Email | varchar(255) | NOT NULL | Store the student’s email |
| 13 | Sport | varchar(255) | NOT NULL | Store the student’s favorite sport |
| 14 | ExUniversity | varchar(100) |  | Store the previous university name |
| 15 | ExEnrollNum | varchar(20) |  | Store the student’s enrollment number in previous university |
| 16 | ExCenter | varchar(50) |  | Store the center name of previous university |
| 17 | ExStream | varchar(50) |  | Store the stream name in previous university |
| 18 | ExField | varchar(50) |  | Store the field name in previous university |
| 19 | ExMarks | decimal |  | Store the student’s marks in previous university |
| 20 | ExOutOfDate | varchar(10) |  | Store the time when student had finished in previous university |
| 21 | ExClass | varchar(10) |  | Store the student’s class in previous university |

#### SPECIALZED SUBJECT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SpeSubjects** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | SubjectID | int  identity(1,1) | PRIMARY KEY | Store the ID of subject |
| 2 | SubjectName | varchar(50) | NOT NULL | Store the Subject Name |
| 3 | FieldID | int | NOT NULL  FOREIGN KEY  REFERENCES  COLUMN “FieldID” ON “Fields” TABLE | Store the Field ID |

#### OPTINAL SUBJECT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OpSubjects** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | SubjectID | int  identity(1,1) | PRIMARY KEY | Store the ID of subject |
| 2 | SubjectName | varchar(50) | NOT NULL | Store the Subject Name |

#### REGISTRATION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Registrations** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | RegistrationID | bigint  identity(1,1) | PRIMARY KEY | Store the ID of the registration |
| 2 | RegNum | varchar(10) | NOT NULL | Store the Registration Number |
| 3 | Image | varchar(max) | NOT NULL | Store the URL of student’s image |
| 4 | SpeSubjectID | int | FOREIGN KEY  REFERENCES  COLUMN “SubjectID” ON “SpeSubjects” TABLE | Store the Specialized Subject ID |
| 5 | OpSubjectID | int | FOREIGN KEY  REFERENCES  COLUMN “SubjectID” ON “OpSubjects” TABLE | Store the Optional Subject ID |
| 6 | EmergencyName | varchar(50) |  | Store the contact name in case of emergency |
| 7 | EmergencAddress | varchar(255) |  | Store the contact address in case of emergency |
| 8 | EmergencyPhone | varchar(12) |  | Store the contact phone number in case of emergency |

#### DEPARTMENT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Departments** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | DepID | int  identity(1,1) | PRIMARY KEY | Store the ID of department |
| 2 | DepName | varchar(50) | NOT NULL | Store the Department Name |
| 3 | Description | Text |  | Store the description of department |
| 4 | Image | varchar(max) |  | Store the URL of image |

#### FACULTY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Facultys** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | FacultyID | int  identity(1,1) | PRIMARY KEY | Store the ID of faculty |
| 2 | FacultyName | varchar(50) | NOT NULL | Store the Faculty Name |
| 3 | DOB | datetime | NOT NULL | Store the faculty’s date of birth |
| 4 | Degree | varchar(10) | NOT NULL | Store the faculty’s degree |
| 4 | DepID | int | NOT NULL  FOREIGN KEY  REFERENCES  COLUMN “DepID” ON “Departments” TABLE | Store the Department ID |
| 6 | Image | varchar(max) |  | Store the URL of faculty’s image |

#### FACILITIES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Facilities** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | FacilityID | int  identity(1,1) | PRIMARY KEY | Store the ID of facility |
| 2 | FacilityName | varchar(50) | NOT NULL | Store the Facility Name |
| 3 | IsActive | bit | NOT NULL | Store the facility’s status is active or not |
| 4 | Image | varchar(max) |  | Store the URL of faculty’s image |

#### FEEDBACK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feedbacks** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | FeedbackID | int  identity(1,1) | PRIMARY KEY | Store the ID of feedback |
| 2 | FirstName | varchar(50) | NOT NULL | Store the First Name of person who send feedback |
| 3 | LastName | varchar(50) | NOT NULL | Store the Last Name of person who send feedback |
| 4 | Email | varchar(255) | NOT NULL | Store the email of person who send feedback |
| 5 | Subject | varchar(max) | NOT NULL | Store the subject of feedback |
| 6 | Message | text | NOT NULL | Store the message of feedback |

#### ACCOUNT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Accounts** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | AccountID | int  identity(1,1) | PRIMARY KEY | Store the ID of account |
| 2 | FullName | varchar(50) | NOT NULL | Store the Full Name of account |
| 3 | Username | varchar(50) | NOT NULL | Store the Username that used to login to admin system |
| 4 | Password | varchar(50) | NOT NULL | Store the Password that used to login to admin system |
| 5 | Role | tinyint | NOT NULL | Store the role number of account |
| 6 | IsActive | bit | NOT NULL | Store the account’s status is active or not |

## TASK SHEET REVIEW II

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Ref. No: 4 | Project Title:  Railway Reservation Manage System | Date of Preparation of Activity Plan | | | |
| No. | Task | Actual Start Date | Actual Days | Team Member Names | Status |
| 01 | Use Cases | 22 Jan , 2022 | 10 | All Members | Completed |
| 02 | Sequence Diagrams | All Members | Completed |
| 03 | Data Flow Diagrams | All Members | Completed |
| 04 | Logical ERD | All Members | Completed |
| 05 | Entity Properties | All Members | Completed |
| 06 | Database Diagram | All Members | Completed |
| 07 | Table Design | All Members | Completed |

|  |  |  |
| --- | --- | --- |
|  | Prepare By: Group 2 | Approved By: Faculty |
| Date: Feb , 2022 | Team Leader  Luu Thanh Sang | Mr. Ngo Phuoc Nguyen |