|  |  |
| --- | --- |
|  | **FPT ACADEMY INTERNATIONAL**  **FPT – APTECH COMPUTER EDUCATION** |

**Centre Name: ACE-HCMC-2-FPT.**

**Address: 590 Cach Mang Thang 8, District 3, Ho Chi Minh City, Viet Nam.**

**Warehouse management**

|  |  |  |
| --- | --- | --- |
| **Supervisor:** | MR. HOANG DUC QUANG | |
| **Semester:** | 4 | |
| **Batch No:** | T1.2009.A0 | |
| **Group No:** | 1 | |
| **Order:** | **Full name** | **Roll No.** |
| 1. | Nguyen Hoang Thien An | Student1268834 |
| 2. | Nguyen Ly Minh Hieu | Student1278909 |
| 3. | Vo Phan Hien | Student1275062 |
| 4. | Pham Ngoc Thien | Student1270702 |

July 2022

**Table of Contents:**

This is to certify that

Mr. Nguyen Hoang Thien An

Mr. Nguyen Ly Minh Hieu

Mr. Vo Phan Hien

Mr. Pham Ngoc Thien



have successfully designed and

developed ****

Submitted by:

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

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



**REVIEW I 5**

**Acknowledgment 6**

**Problem Definition 7**

1. Introduction 7
2. Project Requirement 7
3. Project Designing 7
4. Functional Logic 9

**Task sheet review 1 12**

**REVIEW II 13**

1. Architecture & Design of the Project 14
2. Algorithms - Data Flowchart 16
   1. Main data flow for guest 16
   2. Login Function (Admin only) 18
   3. Logout Function (Admin only) 19
   4. Insert Data Function (Admin only) 20
   5. Update Data Function (Admin only) 21
   6. Delete Data Function (Admin only) 22
   7. Admission Function (Student only) 23
   8. Confirmed Admission (Admin only) 24
   9. Registration Function (Student only) 25
3. Data Flow Diagram 26
   1. Context Diagram 27
   2. Level 0 DFD 28
   3. Level 1 DFD 29
4. Use Case Diagram 33
   1. Administrator Use Case 34
   2. Staff Use Case 35
   3. Guest Use Case 35
5. Entity–Relationship Design 36
   1. Entity–Relationship Diagram 36
   2. Entities and Properties 37

**Task sheet review 2 44**

**REVIEW III 45**

1. Database Design 46
   1. Database Design Diagram 46
   2. Database Structure 47
2. GUI Design – Front-end 55
   1. Home page 55
   2. Courses 58
   3. Admission 59
   4. Department 63
   5. Faculty 64
   6. Facilities 65
   7. Registration 66
   8. Contact/Feedback 69
   9. Admin login 71
3. GUI Design – Back-end 75
   1. Department management 75
   2. Faculty management 86
   3. Facility management 92
   4. Course management 98
   5. Event management 104
   6. Achievement management 110
   7. Branch management 116
   8. Admission management 122
   9. Admin management 125

**Task Sheet Review III 128**

**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. Hoang Duc Quang 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**

1. **Introduction**

Warehouse management involves organizing, managing, and maintaining all the processes that occur in a warehouse, so that they run as smoothly and efficiently as possible. Even though some of these processes are automated and seem error-free, every warehouse operation is prone to mistakes and challenges.

Warehouse management allows you to keep track of your inventory levels, orders, sales, and deliveries. It can also be used to create work orders, bills of materials, and other production-related documents in the manufacturing industry. Warehouse management is used by businesses to avoid product overstock and outages. It’s a tool for organizing inventory data that was previously kept on paper or in spreadsheets.

1. **Project Requirement**

The client had provided all the content details they want to add in the website, including the all information and what links the Dashboard page must contain. Name of the links required in the website for navigation. There should be section in the Dashboard for Sale Order, Incoming List, Inquiry, Configuration (with Warehouse, Location, Supplier, Goods Data, Customer,…) and other sections.

1. **Project Designing**

**DASHBOARD:** Containing charts (income, inventory, goods, sale order,…) and navigation bar to other sections.

**SALE ORDER:** Showing the sale order list

**SALE ORDER DETAIL:** Showing the details of each sale order. In each sale order, showing goods name, price, quantity, shipping quantity, goods status,…

**LOGIN:** There should be separate section for login. The admin person will be able to update the data of the website. After login, people must choose a warehouse to work on.

**USER MANAGEMENT:** User can add management account, add role, set authority, active, change information, password,..

**WAREHOUSE MANAGEMENT:** Manage warehouses of the system, each warehouse has different data flow.

**LOCATION MANAGEMENT:**

1. **Functional Logic**

Loren eps**Hardware / Software Requirements**

1. **Server Requirement**

* Hardware

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| **CPU** | Processor type:  Intel core i5 processor or faster  Processor speed:  Recommended: 2.6 GHz or faster |
| **OS** | Microsoft Windows Server 2012 with IIS |
| **Memory (RAM)** | RAM:  Minimum: 4GB  Recommended: 8 GB or more |
| **Hard Drive** | Free space:  Minimum: 500 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 2017 |

1. **Client Requirement**

* Hardware

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| **CPU** | Processor type:  Pentium IV processor or faster  Processor speed:  Recommended: 2.0 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 |

* Software

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

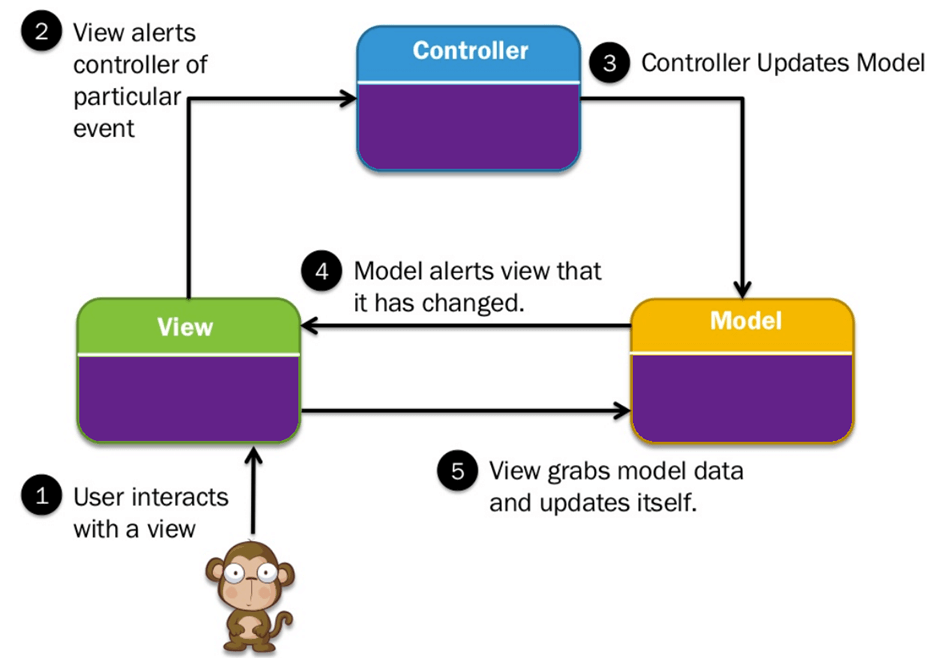
**Task Sheet Review I**

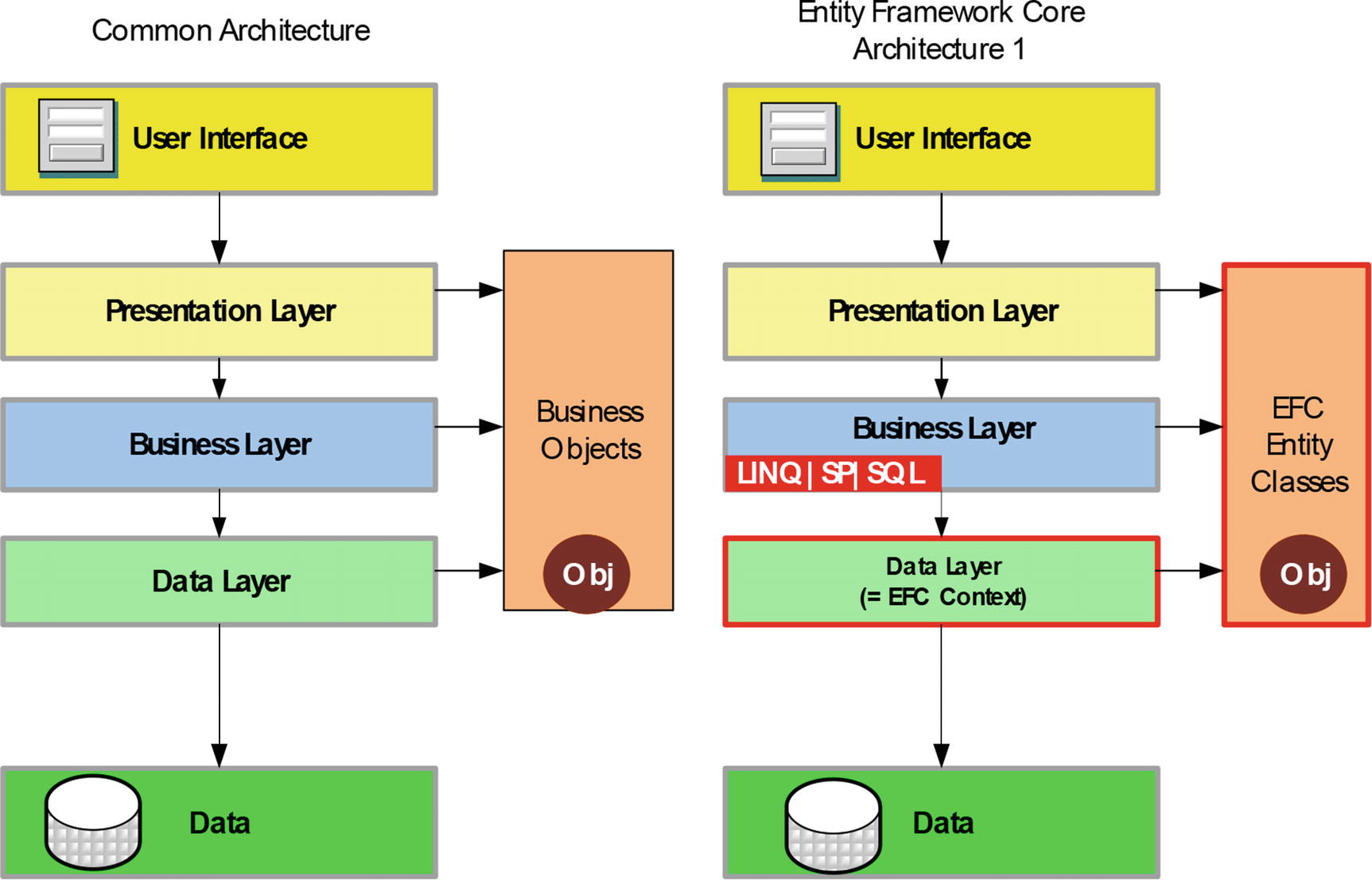
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Ref. No: 4 | Project Title:  Railway Reservation Manage System | Date of Preparation of Activity Plan | | | |
| No. | Task | Actual Start Date | Actual Days | Team Member Names | Status |
| 01 | Acknowledgment | July 1, 2022 | 2 | All 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 1 | Approved By: Faculty |
| Date: Jul 26, 2021 | Team Leader  Nguyen Hoang Thien An | Mr. Hoang Duc Quang |

**REVIEW II**

1. **Architecture & Design of the Project**





Architecture of the Project

## Presentation Tier:

Is the tier in which the users interact with application. Presentation Tier contents Model, View, Controller used to receive a request and response to User.

Technology: Springboot, Thymeleaf, HTML, CSS, JavaScript, Ajax, JQUERY, Bootstrap

## Business Logic Tier:

Is mainly working as the bridge between Data Tier and Presentation Tier. All the Data passes through the Business Tier before passing to the Presentation Tier.

Technology: Spingboot

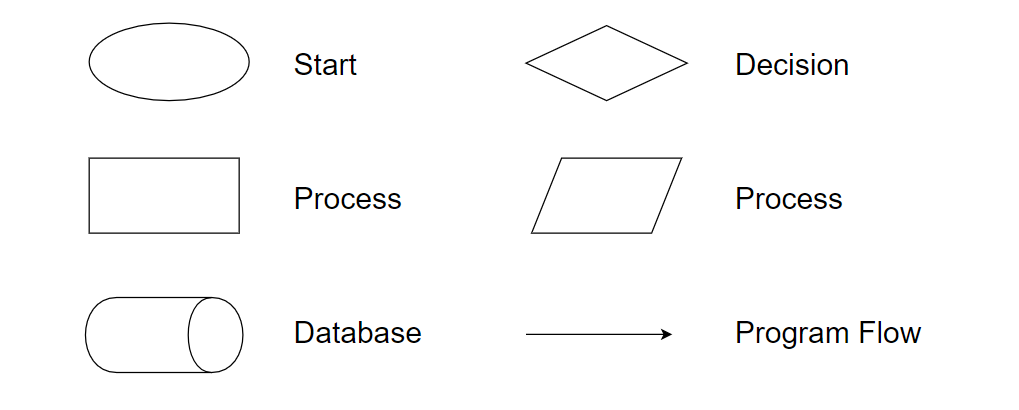
## Data Access Tier:

Is basically the server which stores all the application’s data .Data tier contents Database Tables, Database Views and other means of storing Application Data .

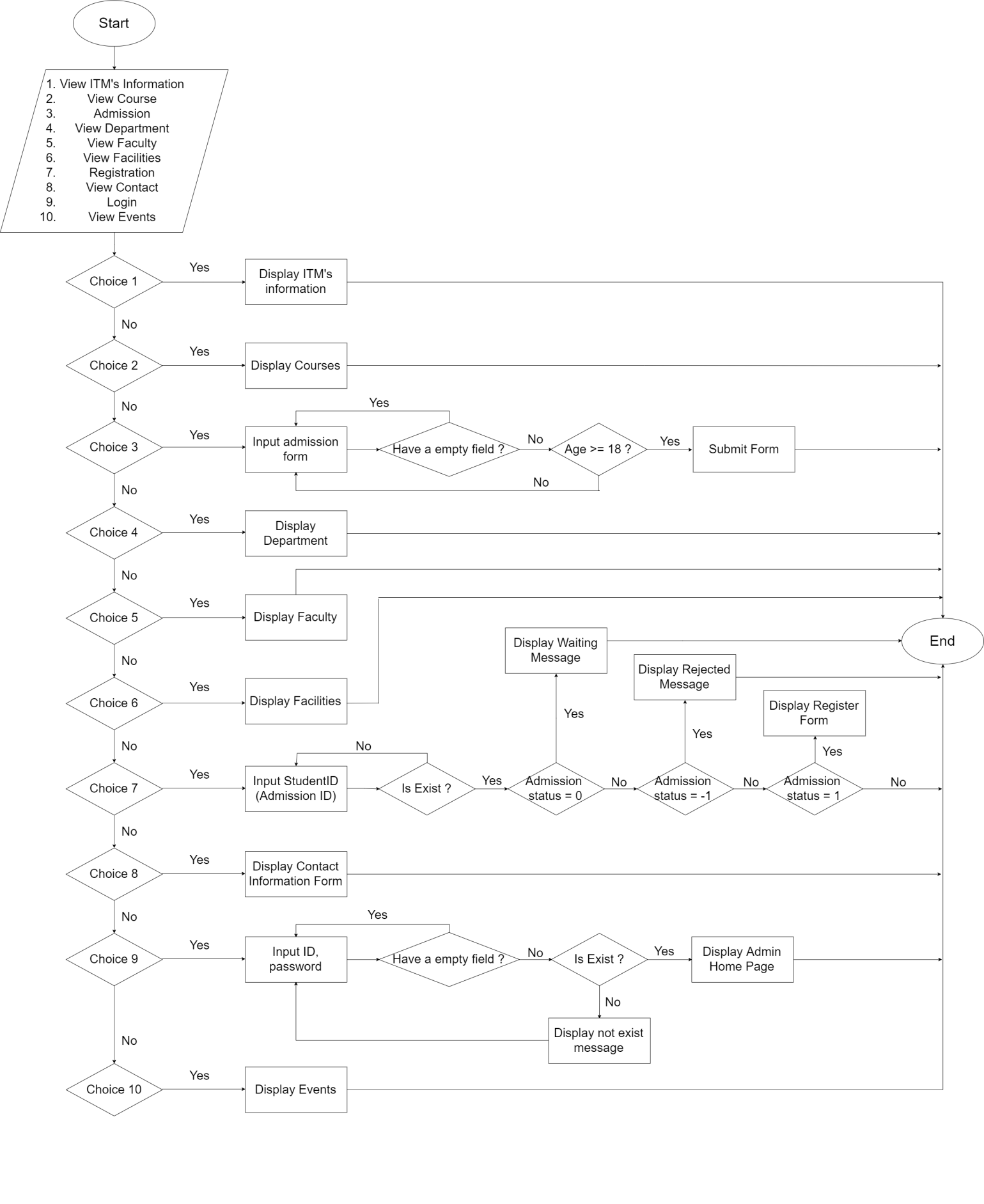
Technology: SQL Server

1. **Algorithms - Data Flowchart**

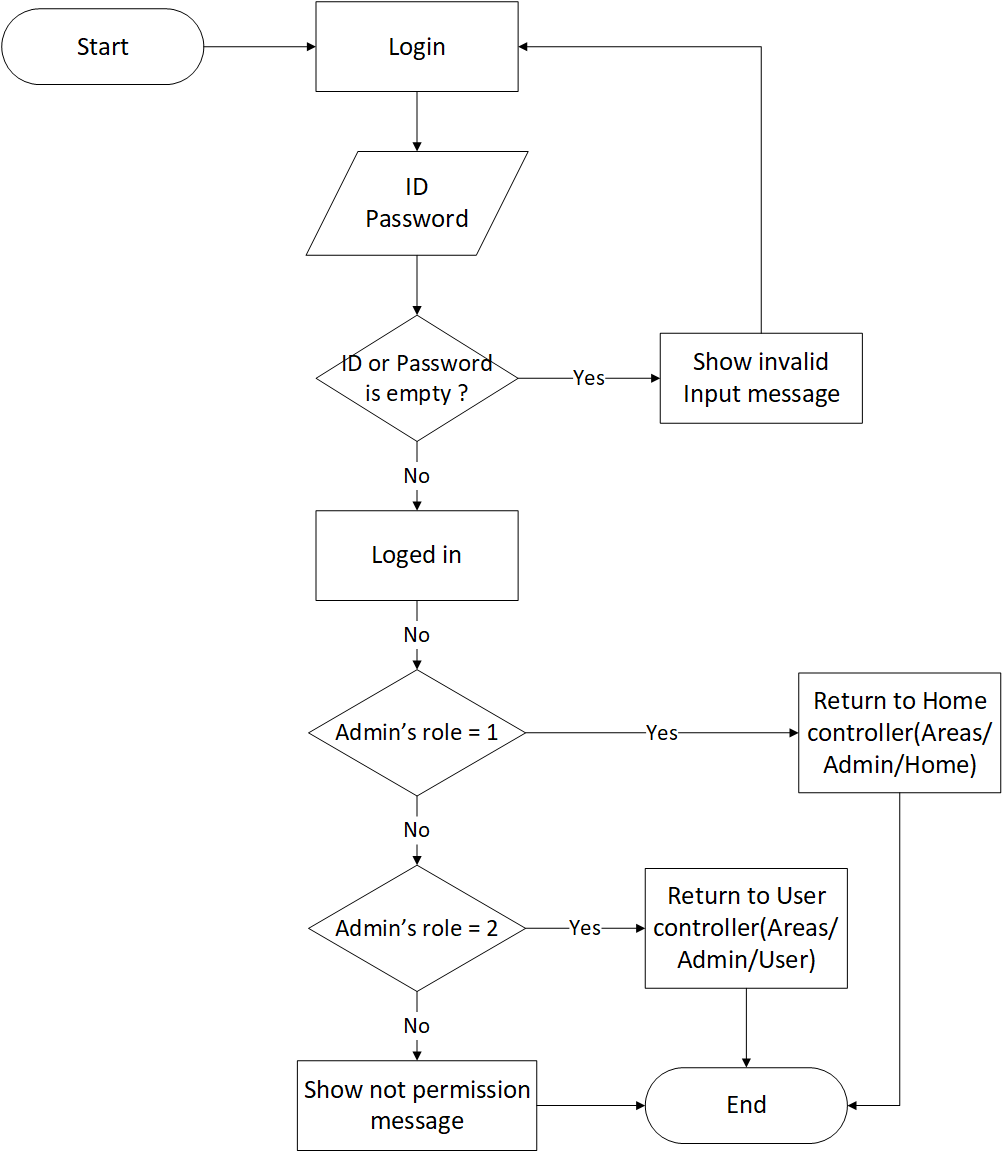
Symbol generates:

****

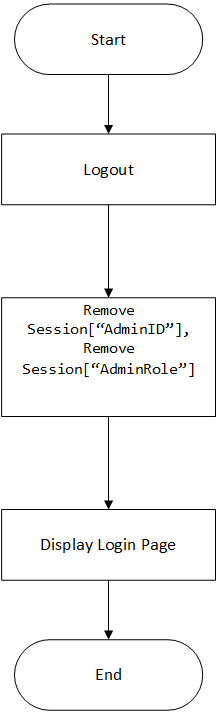
## Main data flow for guest



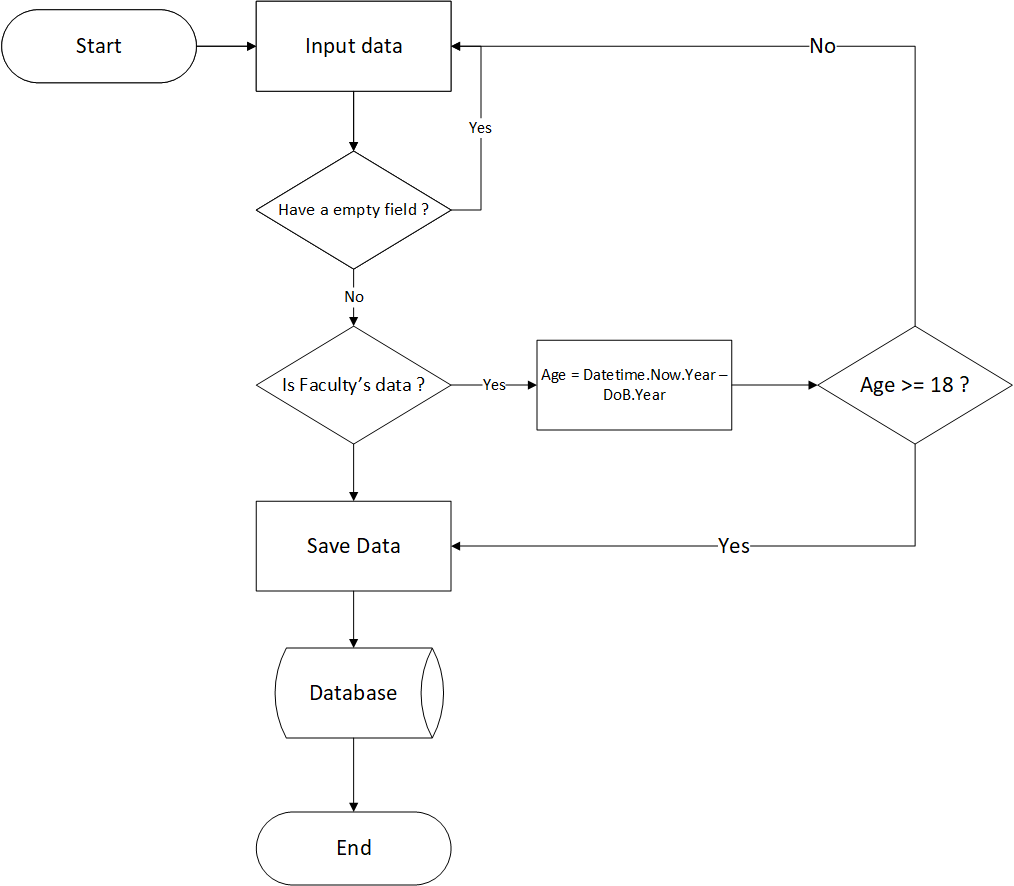
## Login Function (Admin only)



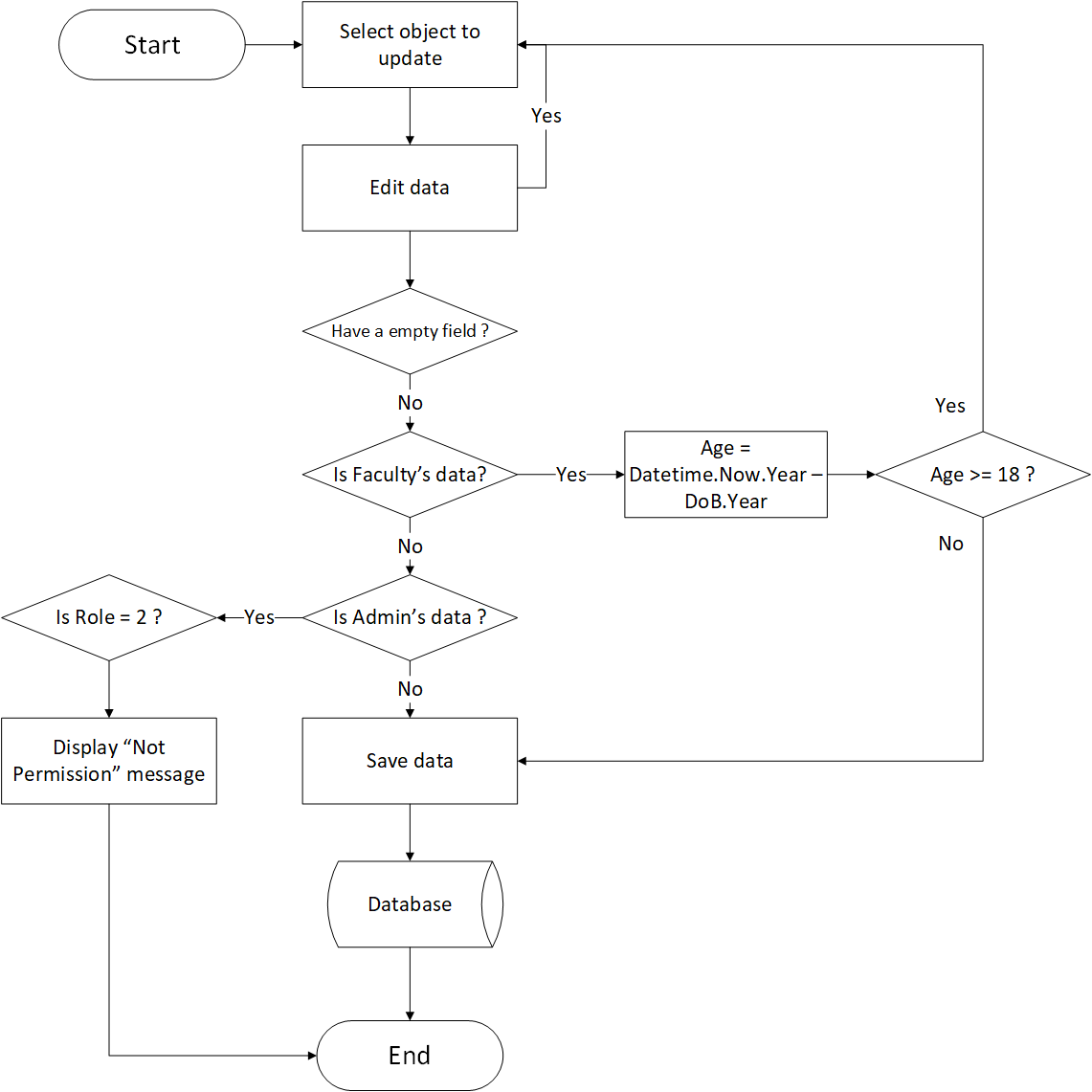
## Logout Function (Admin only)



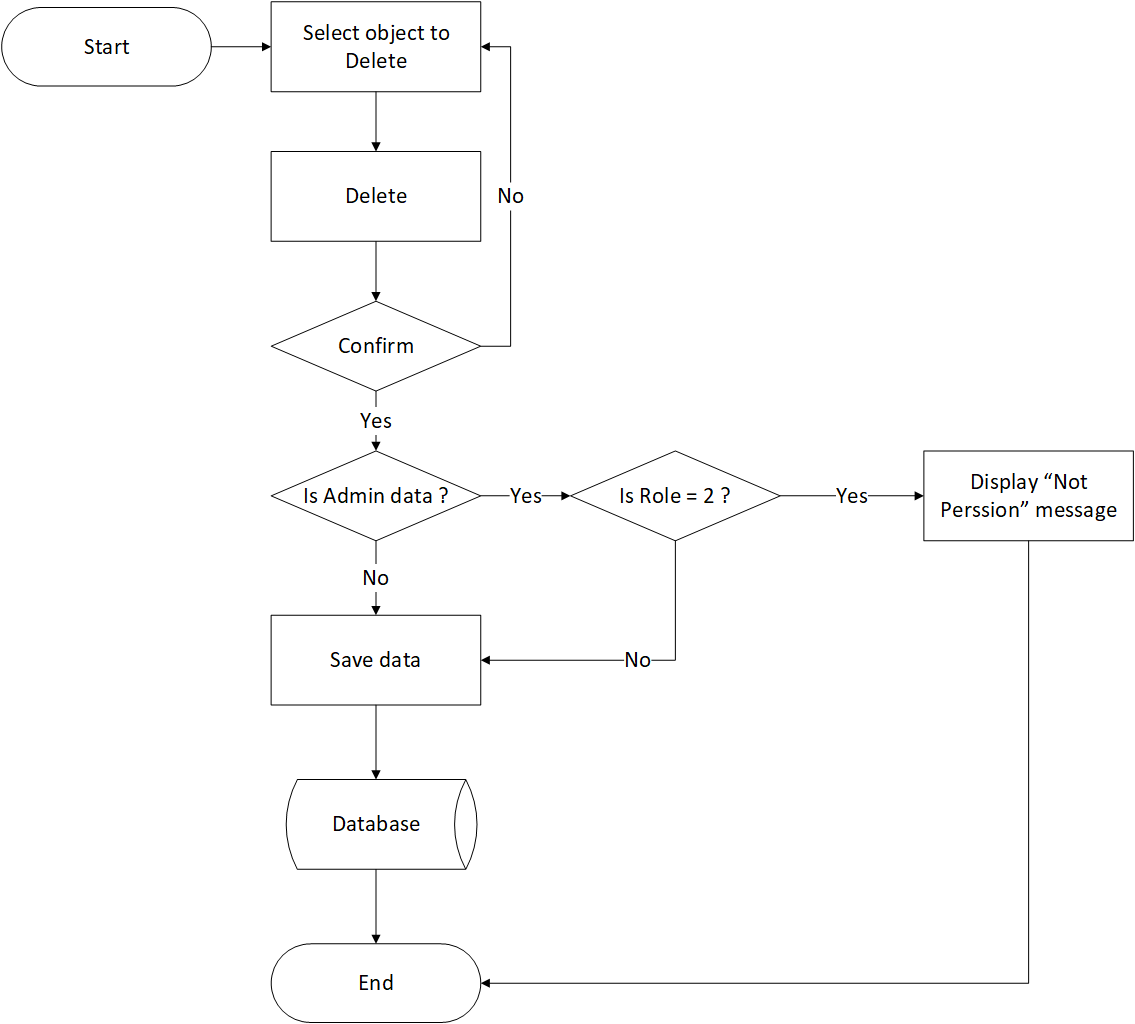
## Insert Data Function (Admin only)



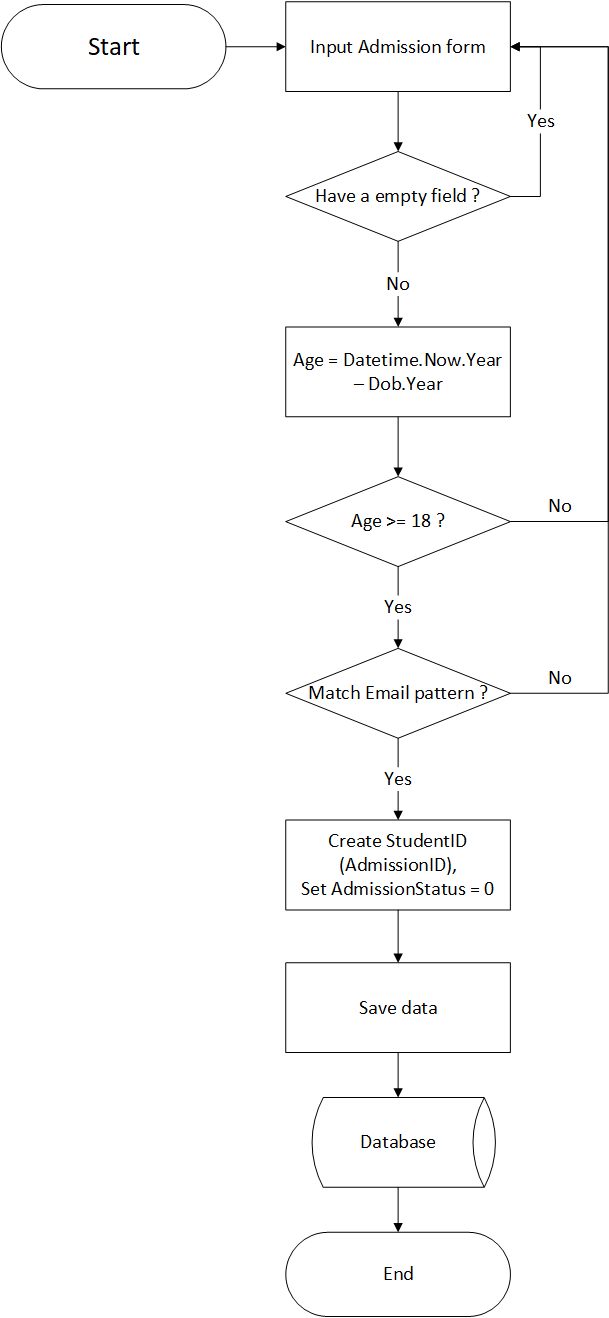
## Update Data Function (Admin only)



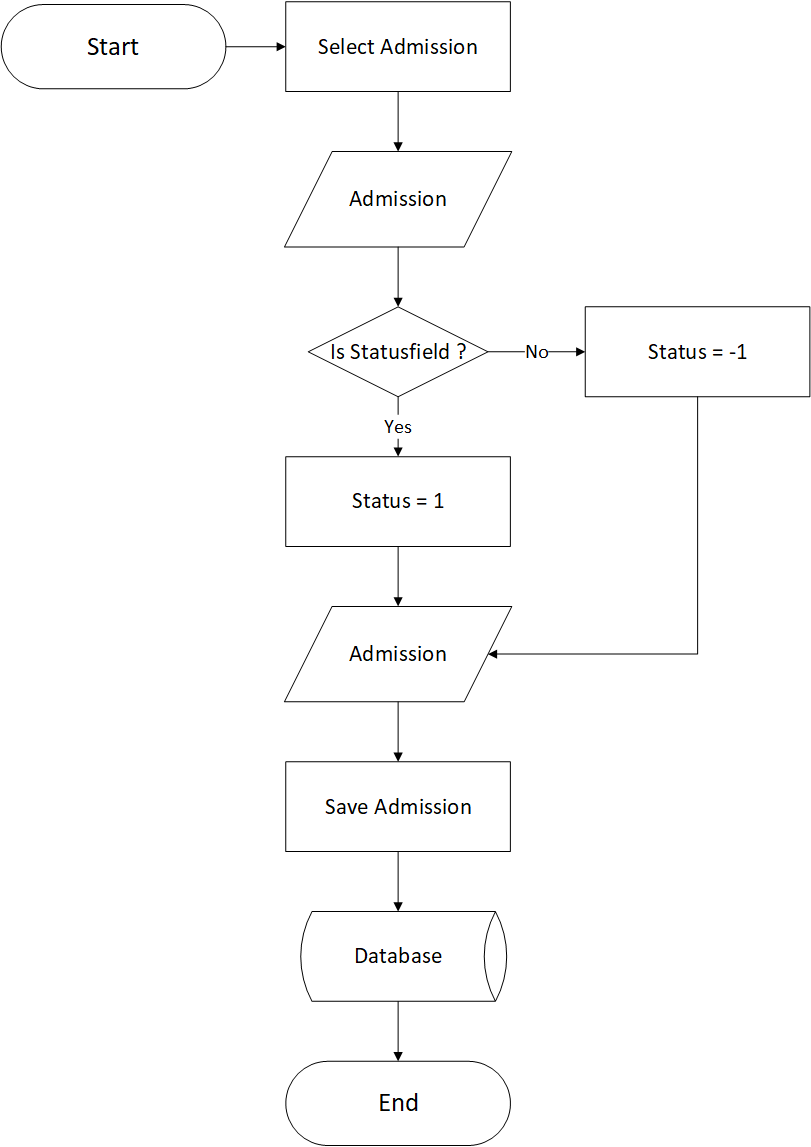
## Delete Data Function (Admin only)



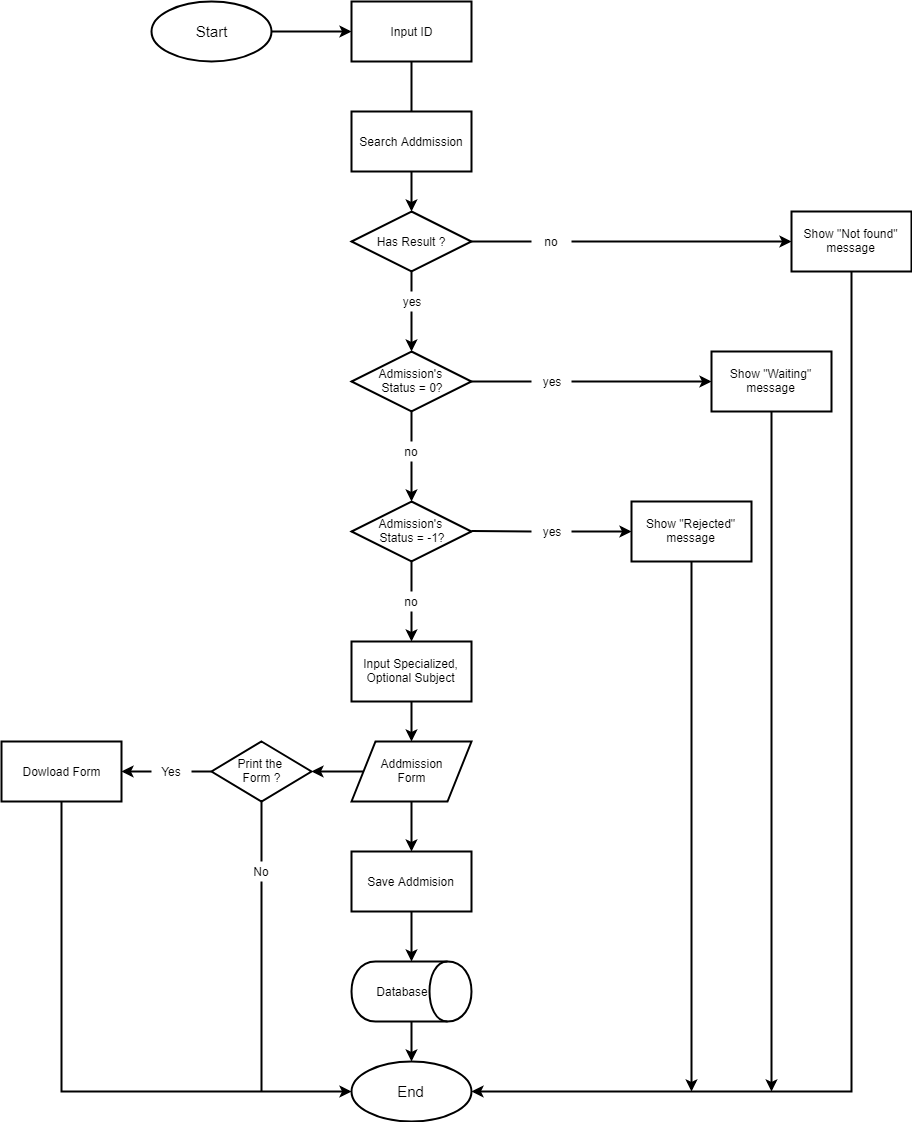
## Admission Function



## Confirmed Admission



## Registration Function

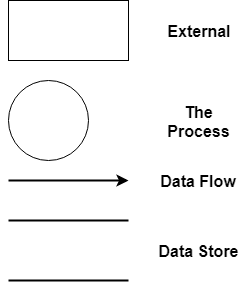


1. **Data Flow Diagram**

**Define:** Data Flows Diagram (DFD) describes the information flow in the system. The next step of system analysis is to consider in detail the information necessary for the implementation for functions discussed above and the one necessary for the improvement of the functions. Modelling tool frequently used for this purpose is DFD. DFD will support 4 main activities:

* **Analysis**: DFD is used to determine requirement of users.
* **Design:** DFD is used to map out plan and illustrate solution to analysis and users while designing a new system.
* **Communication:** One of the strength of DFD is its simplicity and ease to understand to analysts and users.
* **Document:** DFD is used to provide special description of requirement and system design. DFD provide an overview of key functional components of the system but it does not provide any detail on these components. We have to use other tools like database dictionary, process specification to get an idea of which information will be exchanged and how.

**The main components of Context Diagram:**



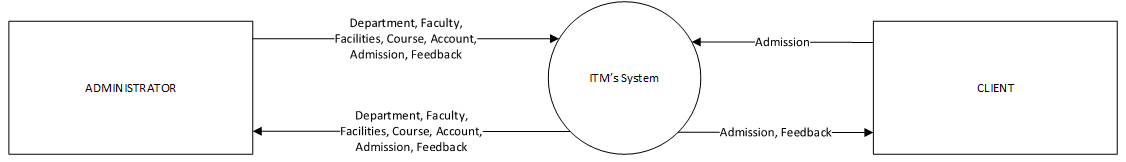
The external factors: External factors can be a person, a group of persons or an organization that are sources of information for the systems and are where system products are transferred to.

The process: Shows the common function of system

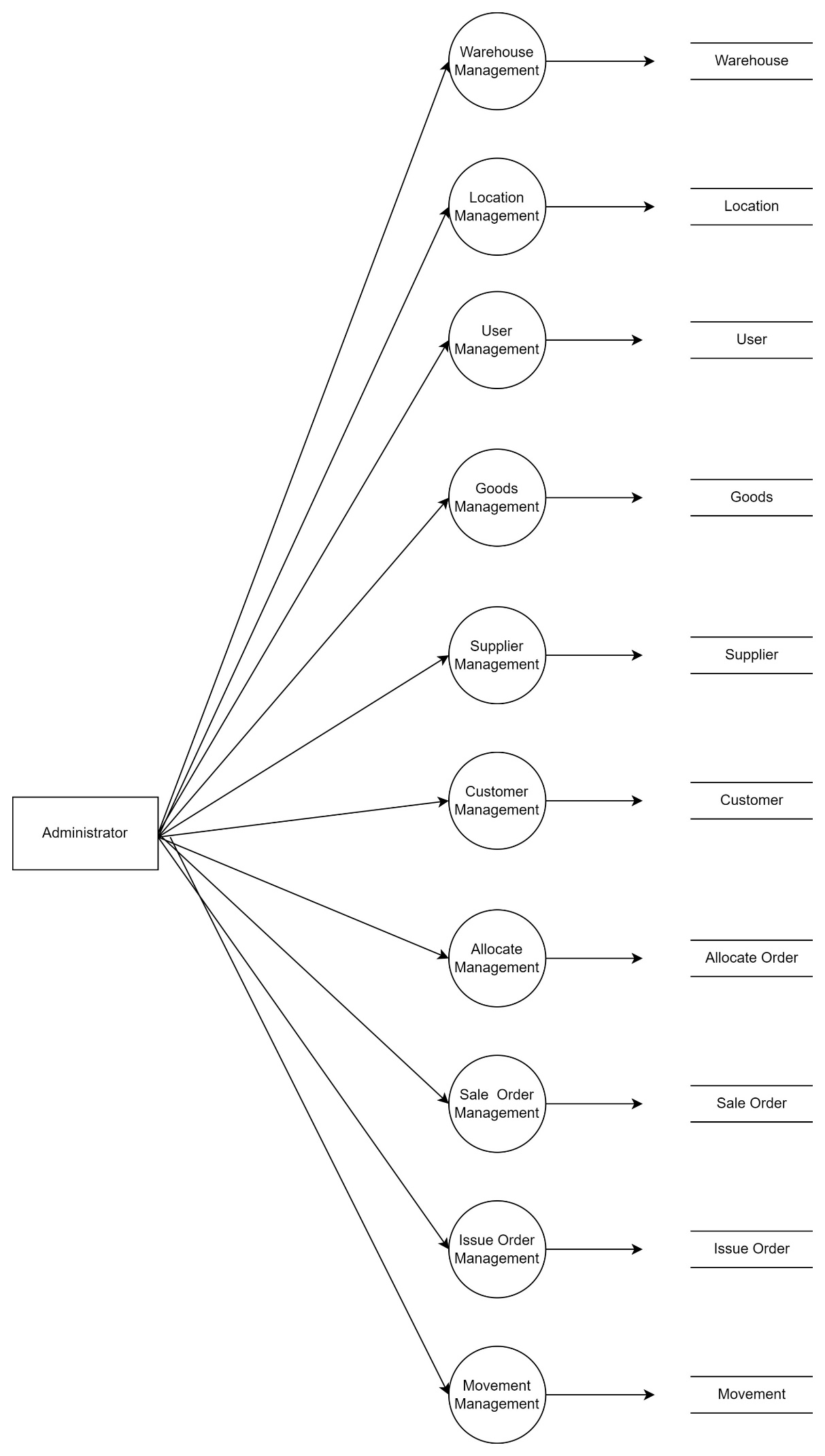
The data flow: Describe the movement of information from one part of the system to another.

The data store: The Data Store is used to model a collection of data packets at rest. A store is represented graphically by two parallel lines. The name of a Data Store that identifies the store is the plural of the name of the packets that are carried by flows into and out of the Data Store.

## Context Diagram:

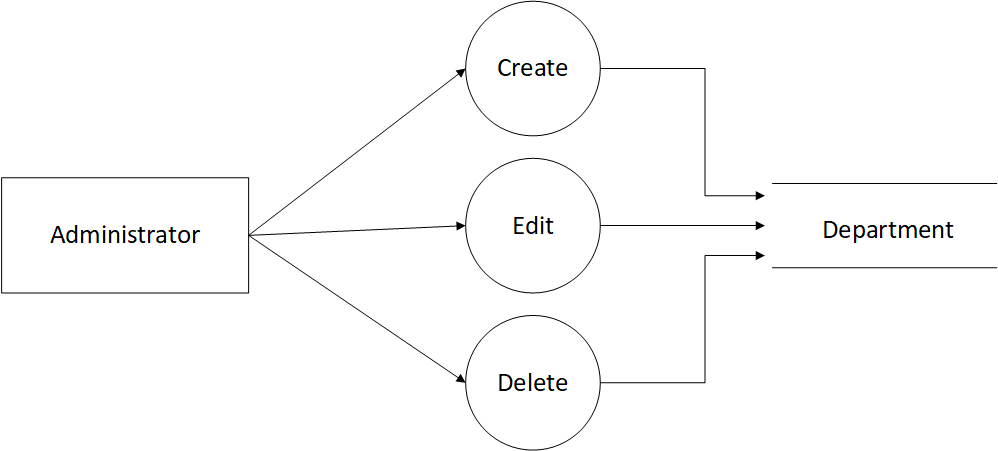
****

## Level 0 DFD

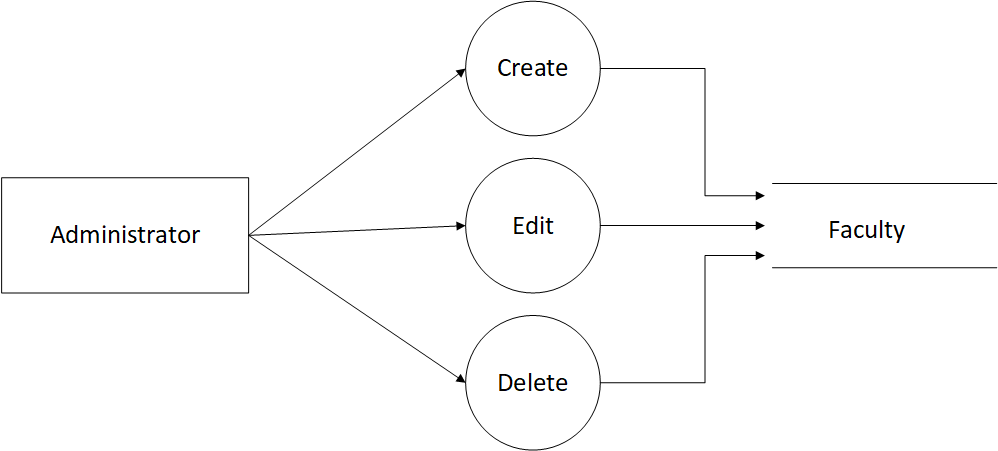


## Level 1 DFD

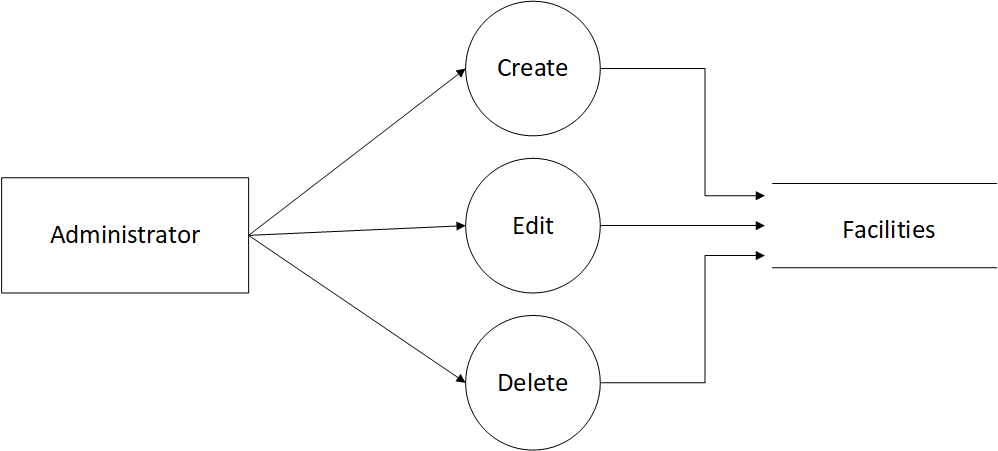
**3.1 Warehouse Management**

****

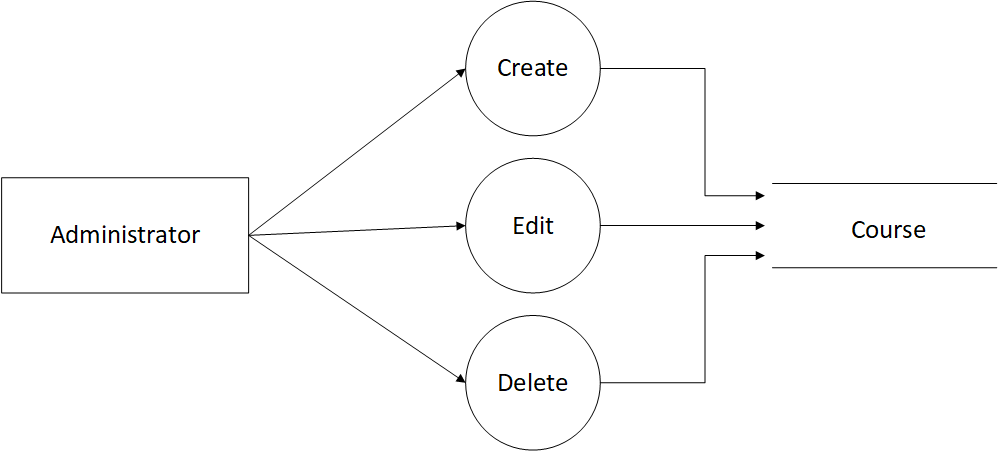
**3.2 Faculty Management**

****

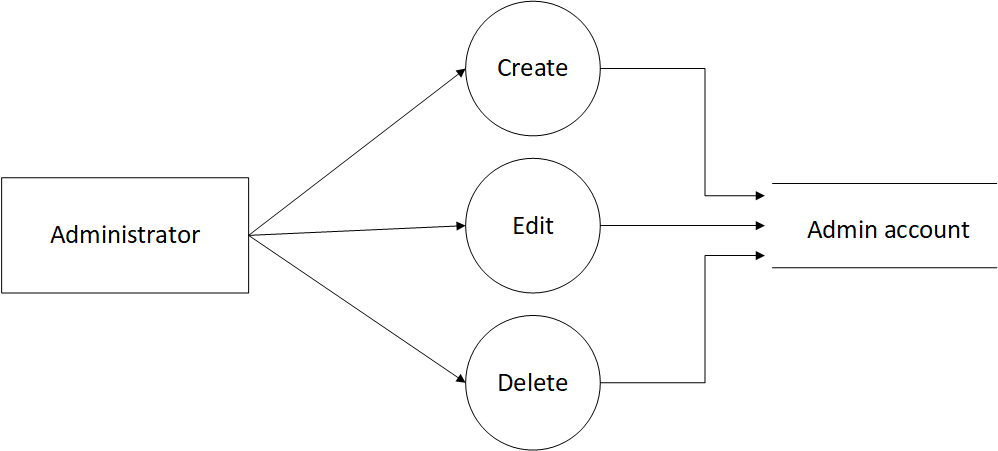
**3.3 Facilities Management**

****

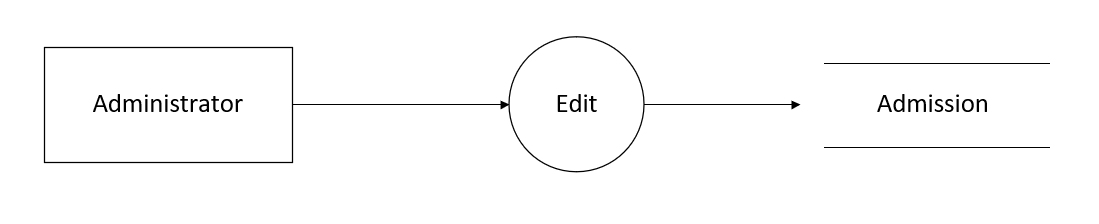
**3.4 Course Management**



**3.5 Admin’s Account Management**

****

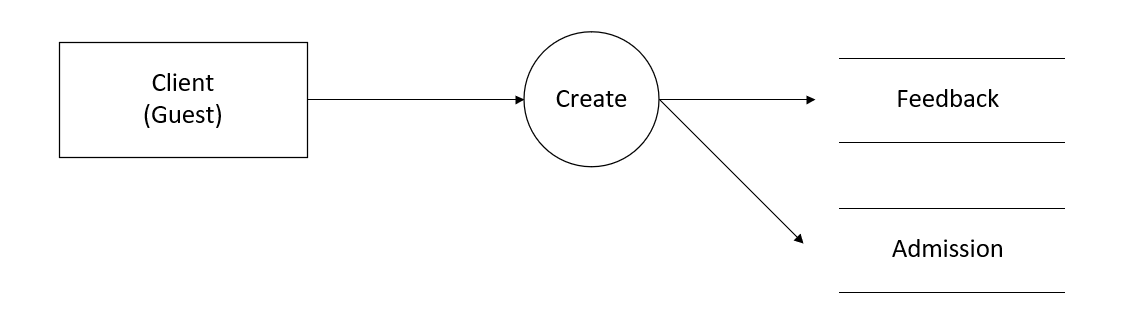
**3.6 Amission Management**

****

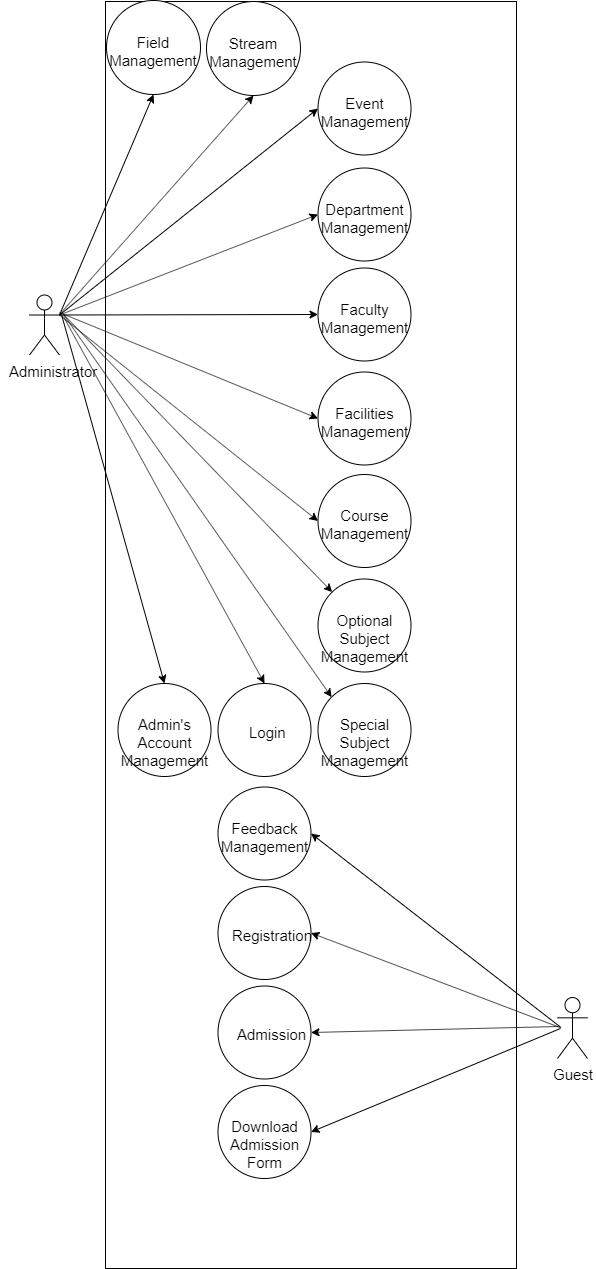
**3.7 Events Management**

****

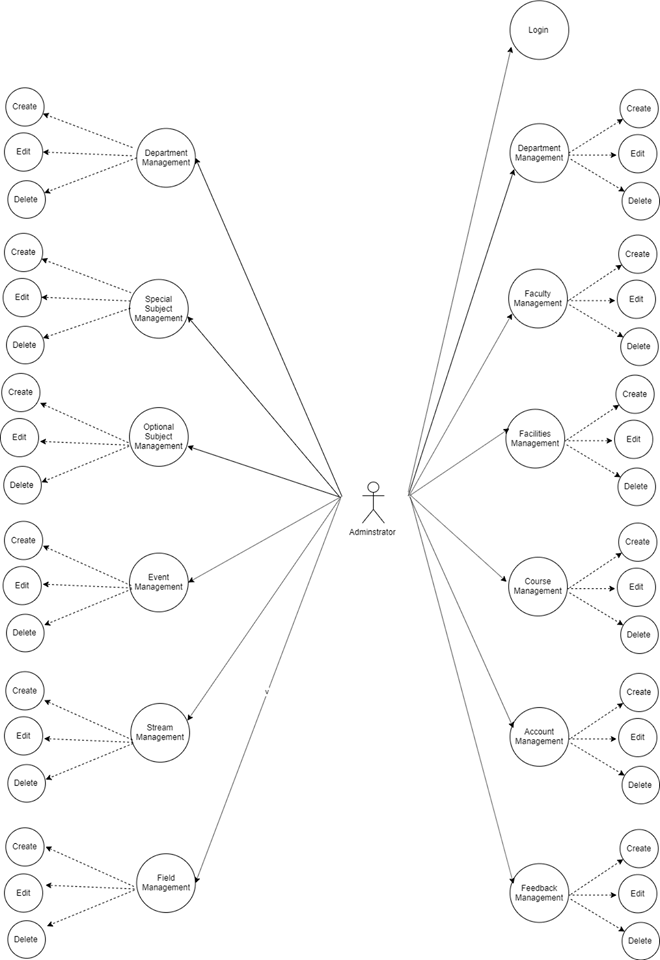
**3.7 Feedback,Admission Management**



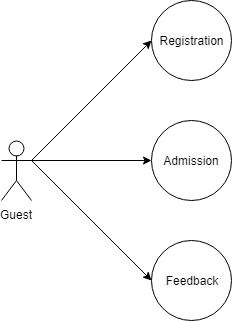
1. **Use Case Diagram**

****

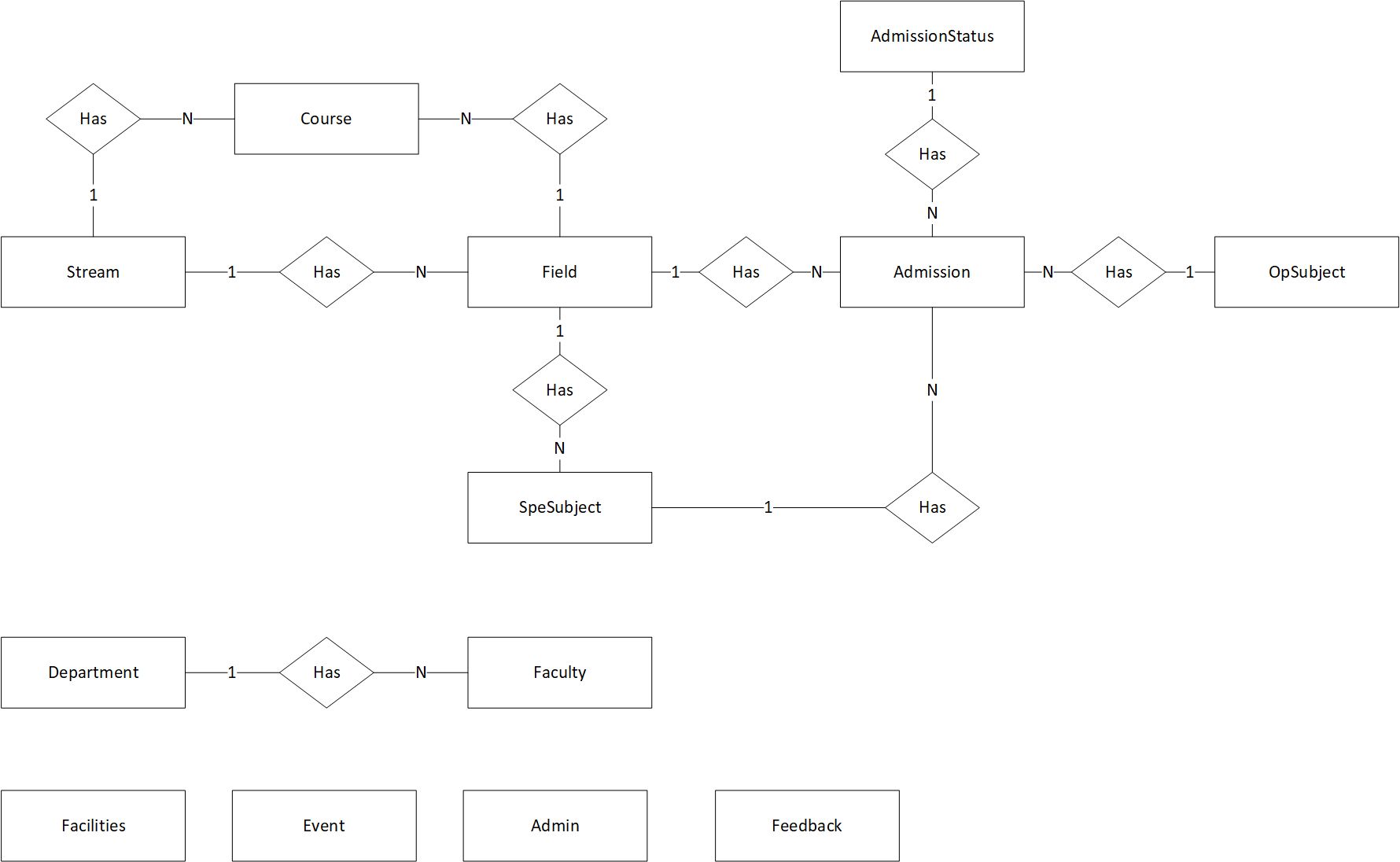
1. Administrator Use Case

****

1. **Guest Use Case**

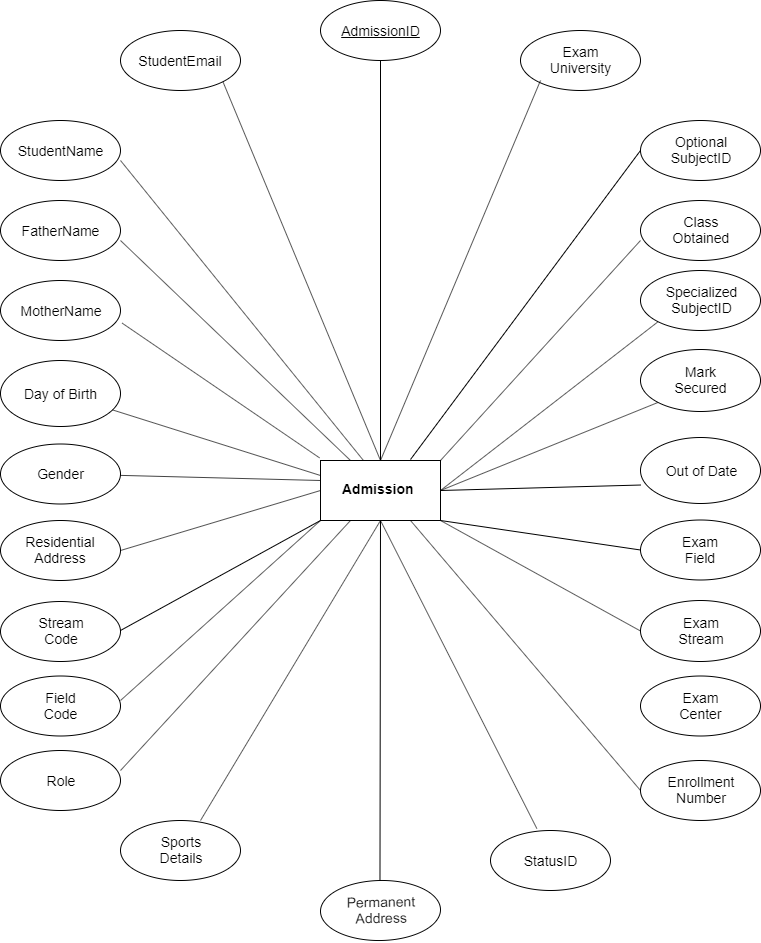
****

1. Entity–Relationship Design
2. **Entity–Relationship Diagram**

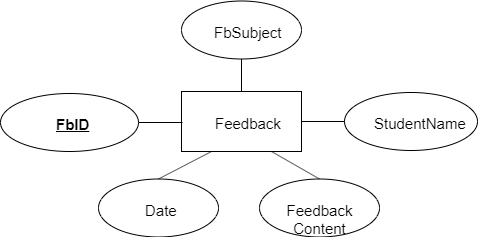
****

1. **Entities and Properties**

**2.1 Admission Entity & Properties**

****

**2.2 Feedback Entity & Properties**



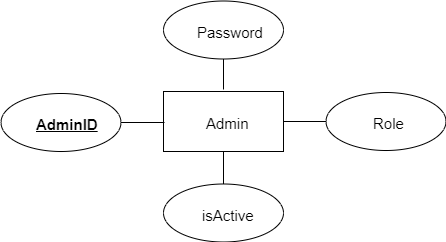
**2.3 Stream Entity & Properties**



**2.4 Field Entity & Properties**



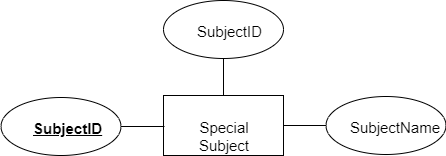
**2.5 Admin Entity & Properties**



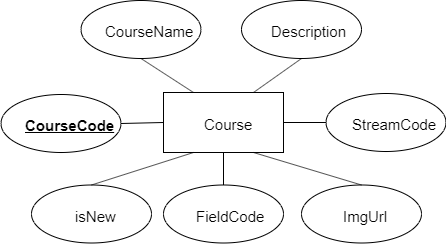
**2.6 Optional Subject Entity & Properties**



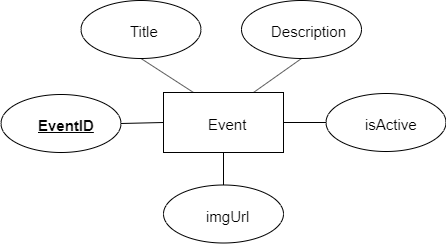
**2.7 Specialized Subject Entity & Properties**



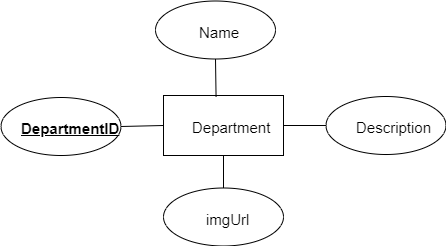
**2.8 Course Entity & Properties**



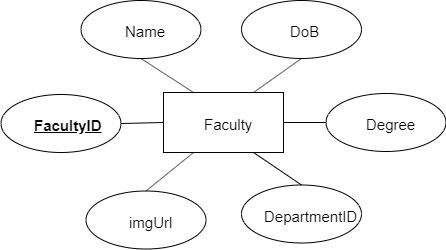
**2.9 Event Entity & Properties**

****

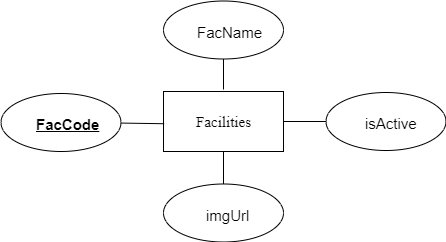
**2.10 Department Entity & Properties**

****

**2.11 Faculty Entity & Properties**

****

**2.12 Facility Entity & Properties**

****

**2.13 Admission Status Entity & Properties**

****

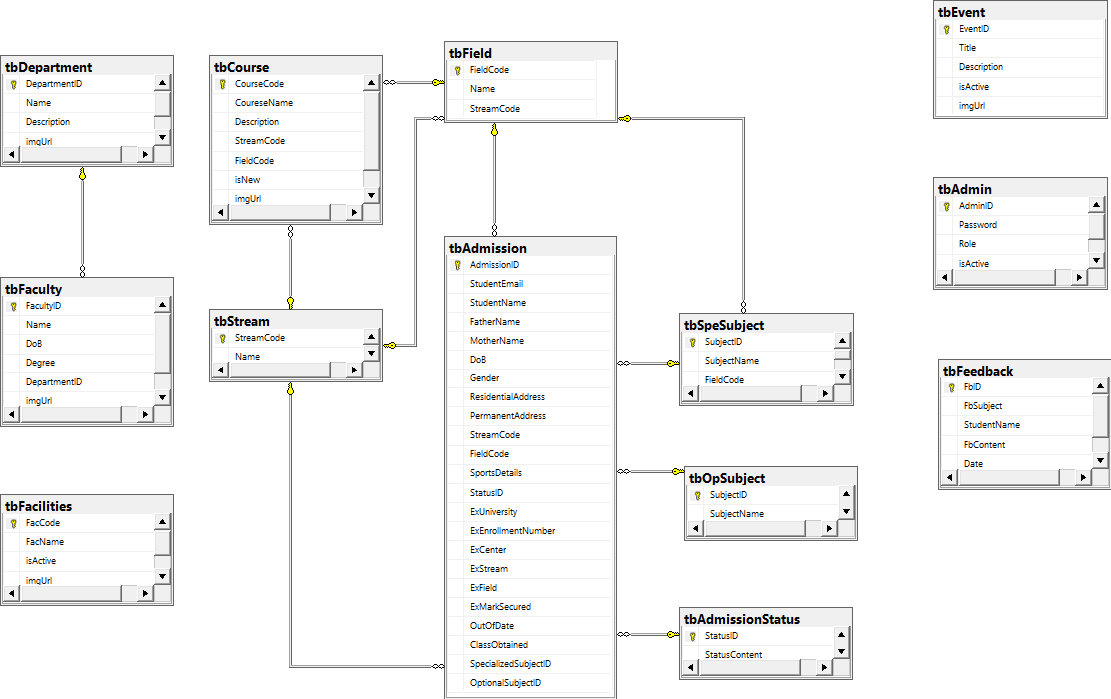
**Task sheet review 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 Name | Status |
| 01 | Architecture & Design of the Project | Nov 8, 2019 | 6 | All Members | Completed |
| 02 | Algorithms - Data Flowchart | Pham The Hien | Completed |
| 03 | Data Flow Diagram | Dinh Thi Kim Ngan | Completed |
| 04 | Use Case Diagram | Vu Duc Thien | Completed |
| 05 | ERD | Nghiem Thi Tam | Completed |

|  |  |  |
| --- | --- | --- |
|  | Prepare By: Group 1 | Approved By: Faculty |
| Date: Nov 8, 2019 | Team Leader  Dinh Thi Kim Ngan | Mr. Ngo Phuoc Nguyen |

**REVIEW III**

1. Database Design
2. Database Design Diagram



1. Database Structure

2.1 Table users

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| id | int | No |  | PK |  | Id of each user |
| username | Varchar(20) | No |  |  |  | Username to Login |
| email | Varchar(50) | No |  |  |  | Email to contact |
| password | Varchar(120) | No |  |  |  | Password for Login |

**2.2 Table users\_roles**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| user\_id | int | No |  | PK |  | User ID |
| role\_id | int | No |  |  |  | Role ID |

**2.3 Table roles**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| id | int | No |  | PK |  | Role ID |
| name | Nvarchar(nax) | No |  |  |  | Name of Role |

**2.4 Table warehouses**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| wh\_code | Varchar(100) | No |  | PK |  | Warehouse code |
| wh\_desc | Varchar(max) | No |  |  |  | Description of warehouse |
| wh\_status | Bit | No |  |  |  | Status of warehouse |
| wh\_cmt | Varchar(max) | No |  |  |  | Comment for warehouse |

**2.5 Table location**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| loc\_code | Varchar(100) | No |  | PK |  | Location Code |
| loc\_desc | Varchar(max) | No |  |  |  | Description of Location |
| loc\_type | Varchar(max) | No |  |  |  | Location Type |
| loc\_status | Bit | No |  |  |  | Enable or Not |
| loc\_cap | float | No |  |  |  | Capable of Location |
| loc\_remain | float | No |  |  |  | Location remain |
| loc\_holding | float | No |  |  |  | Location holding |
| loc\_outgo | float | No |  |  |  |  |
| wh\_code | Varchar(100) | No |  | FK | Warehouse  (wh\_code) | Warehouse Code |
| flammable | Bit | No |  |  |  | Easy to burn or not |

**2.6 Table incomings**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| ic\_id | int | No |  | PK |  | Incoming ID |
| closed | bit | No |  |  |  | Close of Not |
| delivery\_date | Datetime | No |  |  |  | Delivery Date |
| driver | Varchar(255) | No |  |  |  | Driver deliver |
| vehicle | Varchar(255) | No |  |  |  | Vehicle to deliver |
| sup\_code | Varchar(100) |  |  | FK | supplier  (sup\_code) | Supplier Code |

**2.7 Table supplier**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| sup\_code | Varchar(100) | No |  | PK |  | Supplier Code |
| sup\_name | Varchar(max) | No |  |  |  | Name of Supplier |
| sup\_address | Varchar(max) | No |  |  |  | Supplier Address |
| sup\_email | Varchar(max) | no |  |  |  | Email Contact |
| city | Varchar(max) | No |  |  |  | Supplier City |
| active | bit | No |  |  |  | Active or not |
| tax\_code | Varchar(255) | no |  |  |  | Supplier Tax Code |

**2.8 Table goods\_data**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| goods\_no | Varchar(100) | No |  | PK |  | Good No. |
| image | Varchar(max) | No |  |  |  | Image name |
| goods\_name | NVarchar(max) | No |  |  |  | Goods Name |
| price | Int | No |  |  |  | Goods Price |
| active | Bit | No |  |  |  | Active or Not |
| goods\_package | Varchar(100) | No |  |  |  | Package |
| qty\_per\_package | Int | No |  |  |  | Quantity/Package |

**2.9 Table allocation\_request**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| alc\_id | Int | No |  | PK |  | Allocate ID |
| alc\_moved\_qty | int | No |  |  |  | Move Quantity |
| [confirm] | bit | No |  |  |  | Confirm to allocate(Yes/No) |
| movement\_time | Datetime | No |  |  |  | Allocate Date Time |
| pt\_id | Varchar(30) | int |  | **FK** | **goods\_master(pt\_id)** |  |
| loc\_code | Varchar(100) | No |  | **FK** | **location**  **(loc\_code)** | Location Code |
| si\_code | Varchar(200) | Yes |  |  |  |  |

**2.10 Table goods\_master**

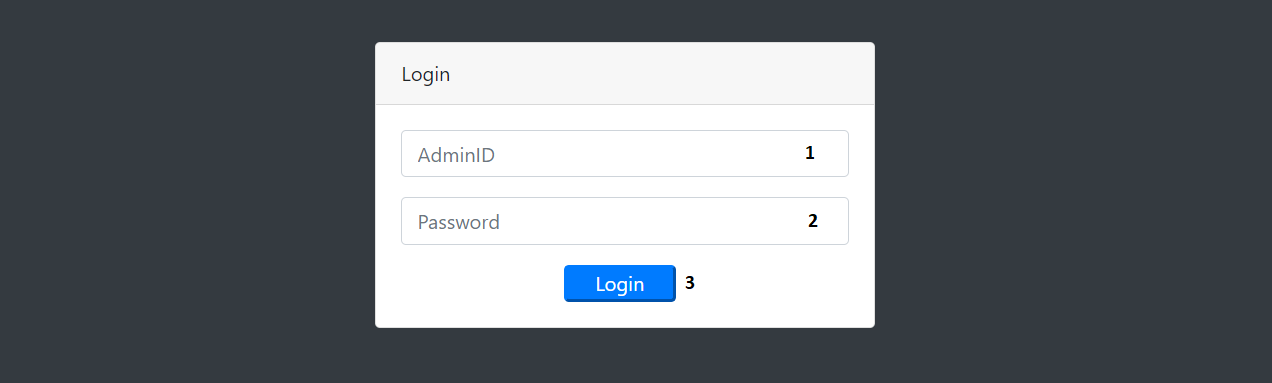
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| pt\_id | int | No |  | PK |  |  |
| goods\_no | Varchar(100) | No |  | **FK** | **goods\_data**  **(goods\_no)** | Goods No. |
| pt\_qty | Float | No |  |  |  |  |
| accepted\_qty | Float | No |  |  |  |  |
| qc | Varchar(100) | No |  |  |  |  |
| pt\_hold | Float | No |  |  |  |  |
| loc\_code | Varchar(100) | No |  | **FK** | **location**  **(loc\_code)** | Location Code |
| pt\_date\_in | DateTime | No |  |  |  |  |
| sup\_code | Varchar(100) | No |  | **FK** | **supplier(sup\_code)** | Supplier Code |
| patch\_no | Varchar(100) | No |  |  |  |  |
| passed | Bit | No |  |  |  |  |
| wh\_code | Varchar(100) | No |  | **FK** | **warehouses(wh\_code)** | Warehouse Code |
| ic\_id | Int | No |  |  |  |  |

**2.11 Table customer**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| cust\_code | Varchar(255) | No |  | PK |  | Customer Code |
| active | Bit | No |  |  |  | Active or Not |
| address | Varchar(255) | No |  |  |  | Customer Address |
| city | Varchar(255) | No |  |  |  | Customer City |
| country | Varchar(255) | No |  |  |  | Customer Country |
| cust\_name | Varchar(255) | No |  |  |  | Customer Name |
| email | Varchar(255) | No |  |  |  | Email Contact |
| note | Varchar(255) | No |  |  |  | Note |
| phone | Varchar(255) | No |  |  |  | Phone Contact |
| short\_name | Varchar(255) | No |  |  |  | Short Name |
| tax\_code | Varchar(255) | No |  |  |  | Tax Code |

GUI Design – Front-end:

1. Admin login:





|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Login page** | | | | | | |
| No | Name | Type | Validation | Event | Description | Status |
| 1 | ID | Text box | Not null | Input | Input admin id | Enable |
| 2 | Password | Text box | Not null | Input | Input password | Enable |
| 3 | btnSubmit | Button |  | Click | Submit for login | Enable |
| 4 | Department | HyperLink |  | Click | URL to Departments action (Areas/Admin/Home controller) | Enable |
| 5 | Event | HyperLink |  | Click | URL to Event action (Areas/Admin/Home controller) | Enable |
| 6 | Facilities | HyperLink |  | Click | URL to Facilities action (Areas/Admin/Home controller) | Enable |
| 7 | Faculty | HyperLink |  | Click | URL to Faculties action (Areas/Admin/Home controller) | Enable |
| 8 | Feedback | HyperLink |  | Click | URL to Feedback action (Areas/Admin/Home controller) | Enable |
| 9 | Course | HyperLink |  | Click | URL to Course action (Areas/Admin/Home controller) | Enable |
| 10 | Field | HyperLink |  | Click | URL to Field action (Areas/Admin/Home controller) | Enable |
| 11 | Stream | HyperLink |  | Click | URL to Stream action (Areas/Admin/Home controller) | Enable |
| 12 | Special Subject | HyperLink |  | Click | URL to SpeSubjects action (Areas/Admin/Home controller) | Enable |
| 13 | Optional Subject | HyperLink |  | Click | URL to OpSubjects action (Areas/Admin/Home controller) | Enable |
| 14 | Admission List | HyperLink |  | Click | URL to AdmissionList action (Areas/Admin/Home controller) | Enable |
| 15 | Admin’s Account List | HyperLink |  | Click | URL to Index action (Areas/Admin/User controller) | Enable |

* Some code review:
* Login:

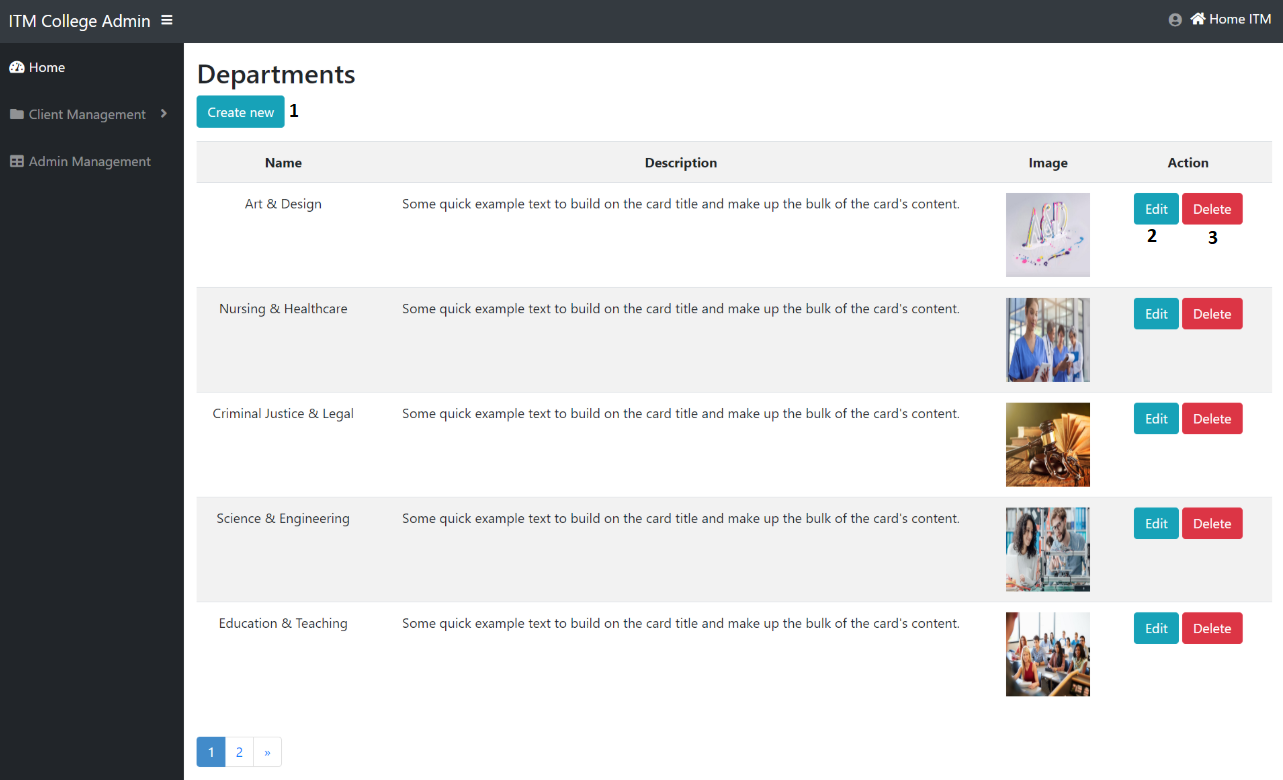


* Logout:



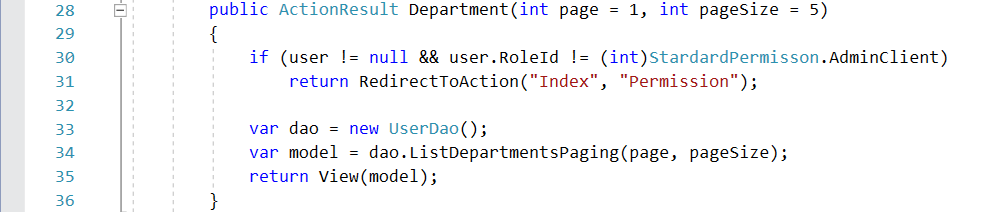
1. GUI Design – Back-end:
2. Department management

**a. Department Post List**

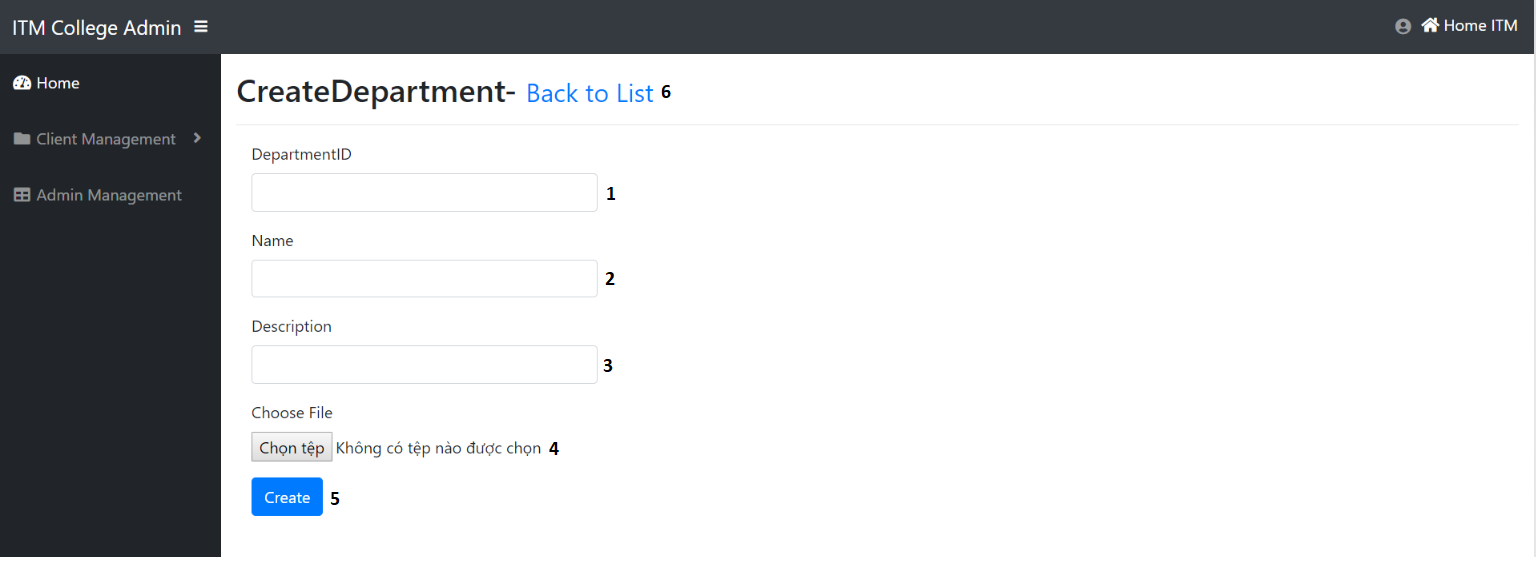
****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Department Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | Create | Hyberlink |  | Click | Create new Department Post | Enable |
| 3 | btnEdit | Button |  | Click | Edit Department Post | Enable |
| 4 | btnDelete | Button |  | Click | Delete Department Post | Enable |

* Some code review:
* Get department list:



**b. Create Department Post**

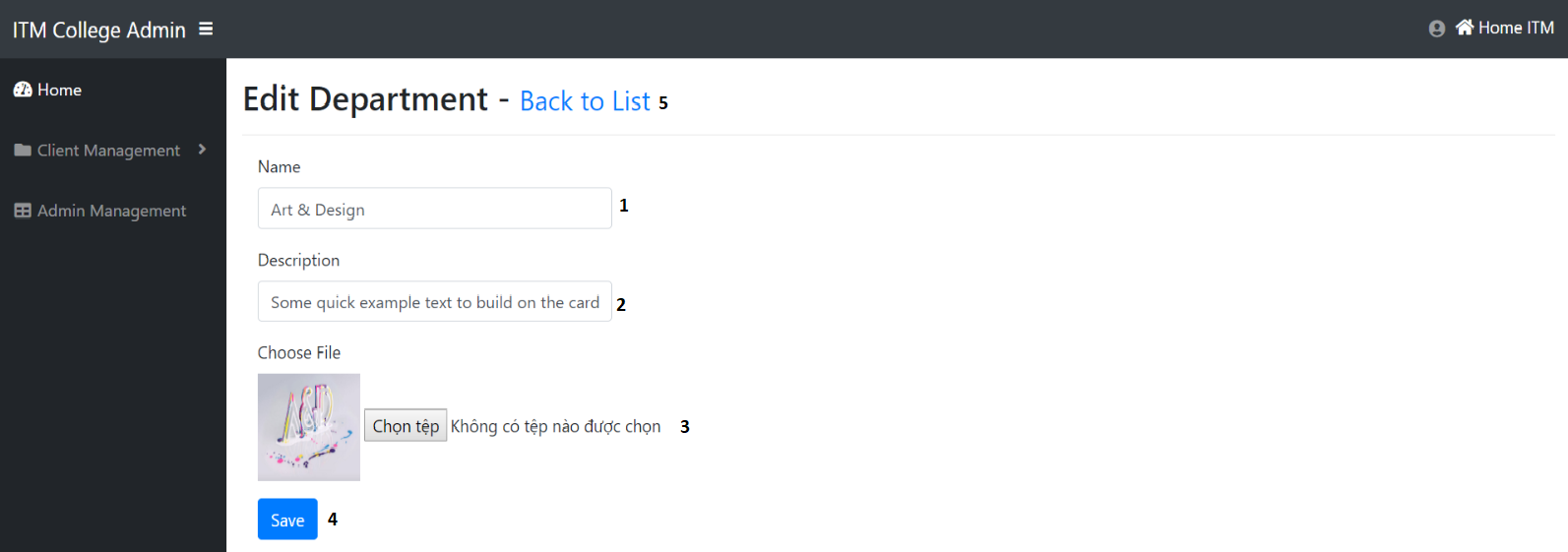
****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Create Department Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | DepartmentID | TextField | 0 <= Length <= 30 | Input | Input Department ID | Enable |
| 2 | Name | TextField | 0 <= Length <= 50 | Input | Input Department Name | Enable |
| 3 | Description | TextField |  | Input | Input Description | Enable |
| 4 | Choose File | File | Required | Click | Path of Image | Enable |
| 5 | btnCreate | Button |  | Click | Submit form Create | Enable |
| 6 | Back to list | Hyberlink |  | Click | View Department list | Enable |

* Some code review:
* Create Deparment:



**c. Edit Department Post**

****

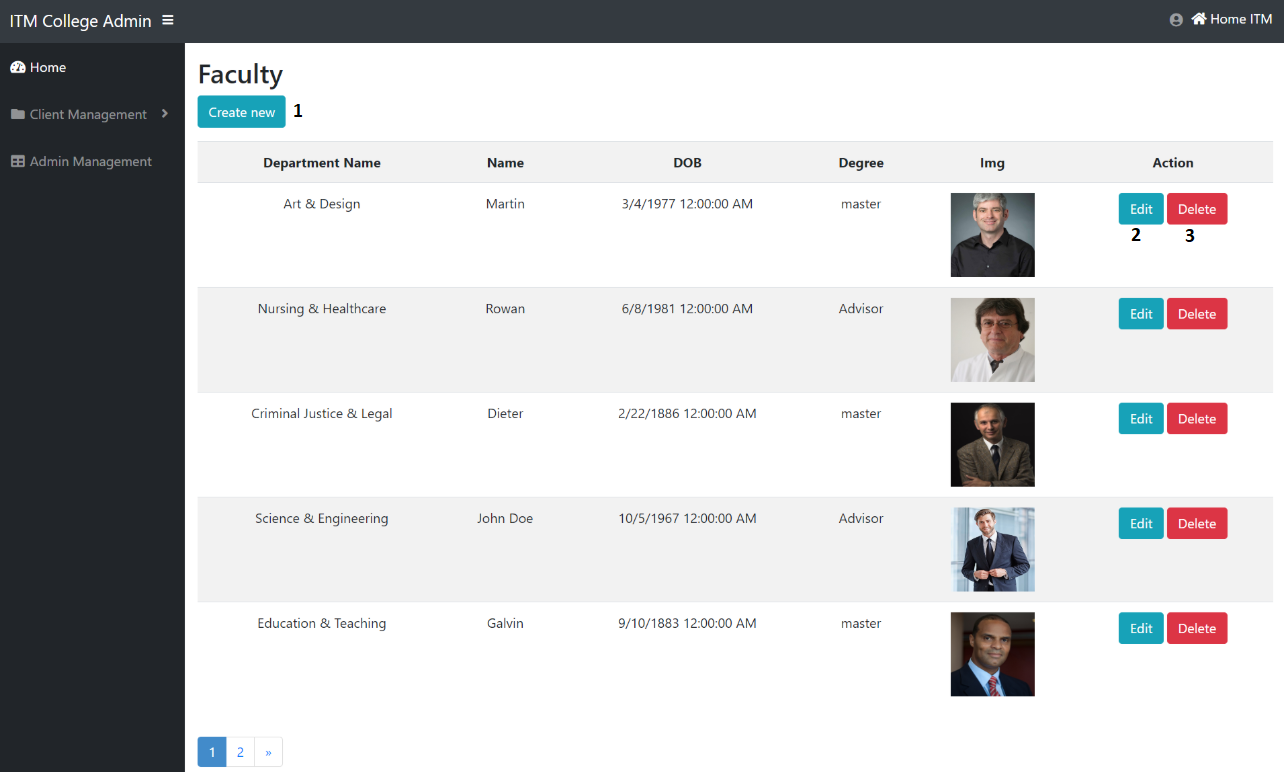
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Edit Department Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | Name | TextField | 0 <= Length <= 50 | Input | Input Department Name | Enable |
| 2 | Description | TextField |  | Input | Input Description | Enable |
| 3 | Choose File | File | Required | Click | Path of Image | Enable |
| 4 | btnSave | Button |  | Click | Submit form Edit | Enable |
| 5 | Back to list | Hyberlink |  | Click | View Department List | Enable |

* Some code review:
* Edit Department:



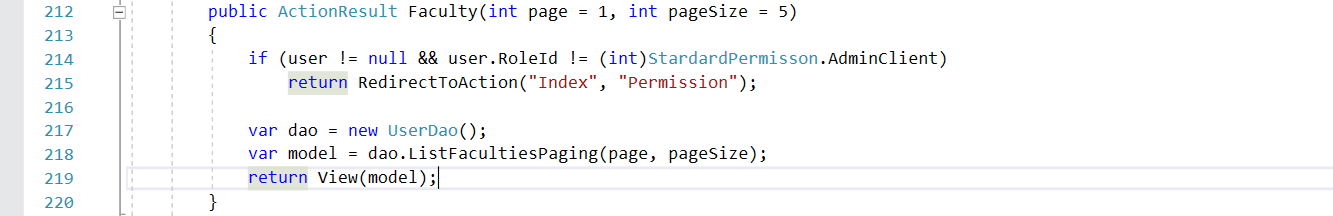
1. Faculty management

**a. Faculty Post List**

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Faculty Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | Create | Hyberlink |  | Click | Create new Faculty Post | Enable |
| 3 | btnEdit | Button |  | Click | Edit Faculty Post | Enable |
| 4 | btnDelete | Button |  | Click | Delete Faculty Post | Enable |

* Some code review:
* Get Faculty list:

****

**b. Create Faculty Post**

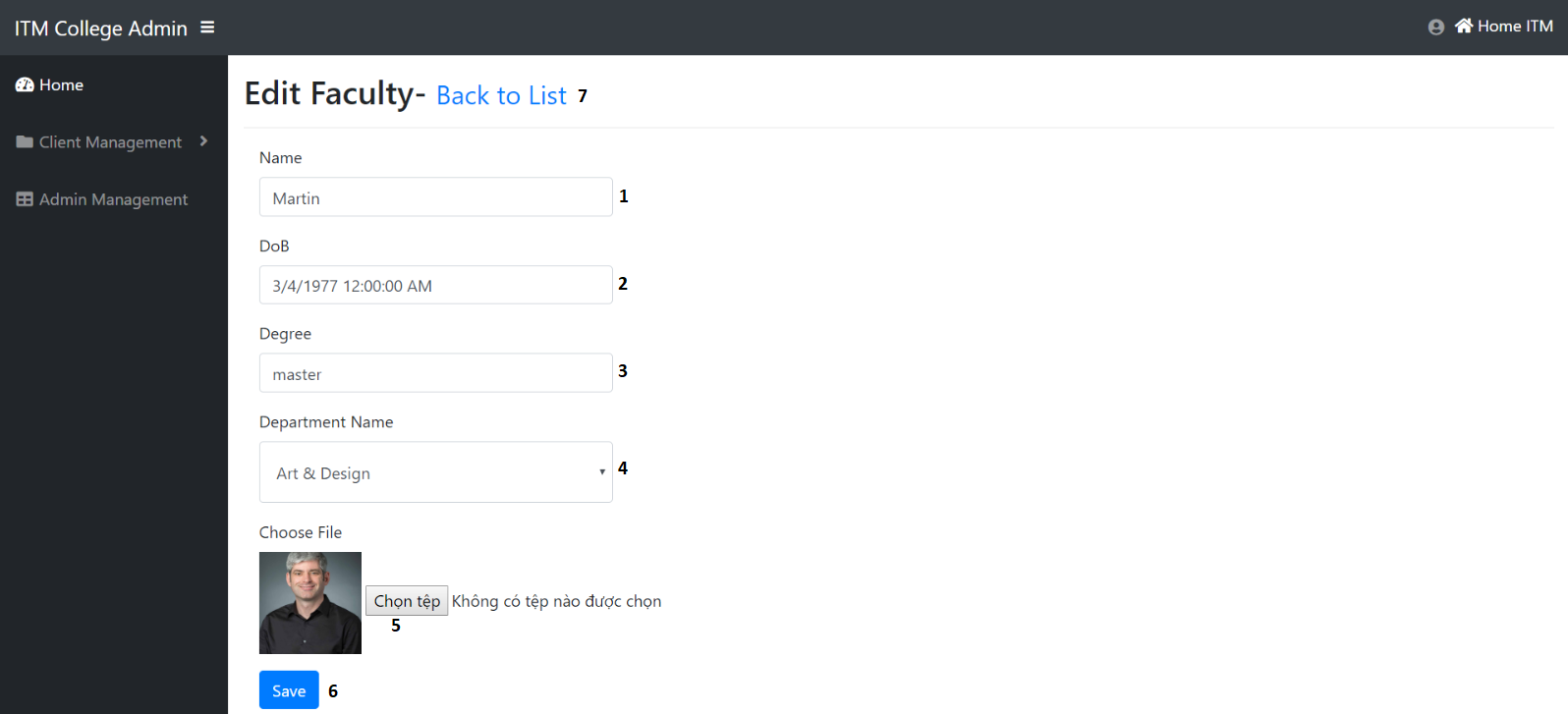
****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Create Faculty Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | FaculID | TextField | 0 <= Length <= 30 | Input | Input Faculty ID | Enable |
| 2 | Name | TextField | 0 <= Length <= 50 | Input | Input Faculty Name | Enable |
| 3 | DoB | TextField | Datetime | Input | Input Faculty Day of Birth | Enable |
| 4 | Degree | TextField | 0 <= Length <= 20 | Click | Input Degree of Faculty | Enable |
| 5 | Department | Dropdown |  | Click | Choose Department | Enable |
| 6 | Choose File | File | Required | Click | Path of Image | Enable |
| 7 | btnCreate | Button |  | Click | Submit form Create | Enable |
| 8 | Back to list | Hyberlink |  | Click | URL to faculty list | Enable |

* Some code review:
* Create Faculty:

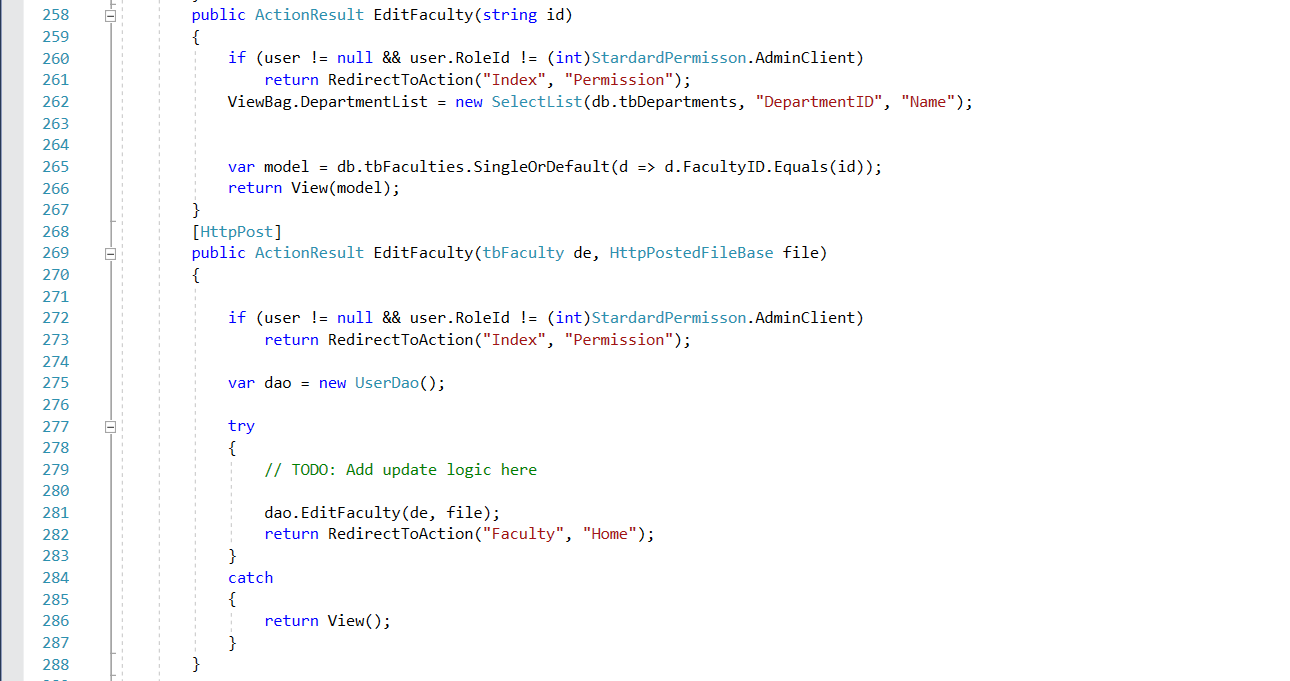


**c. Edit Faculty Post**

****

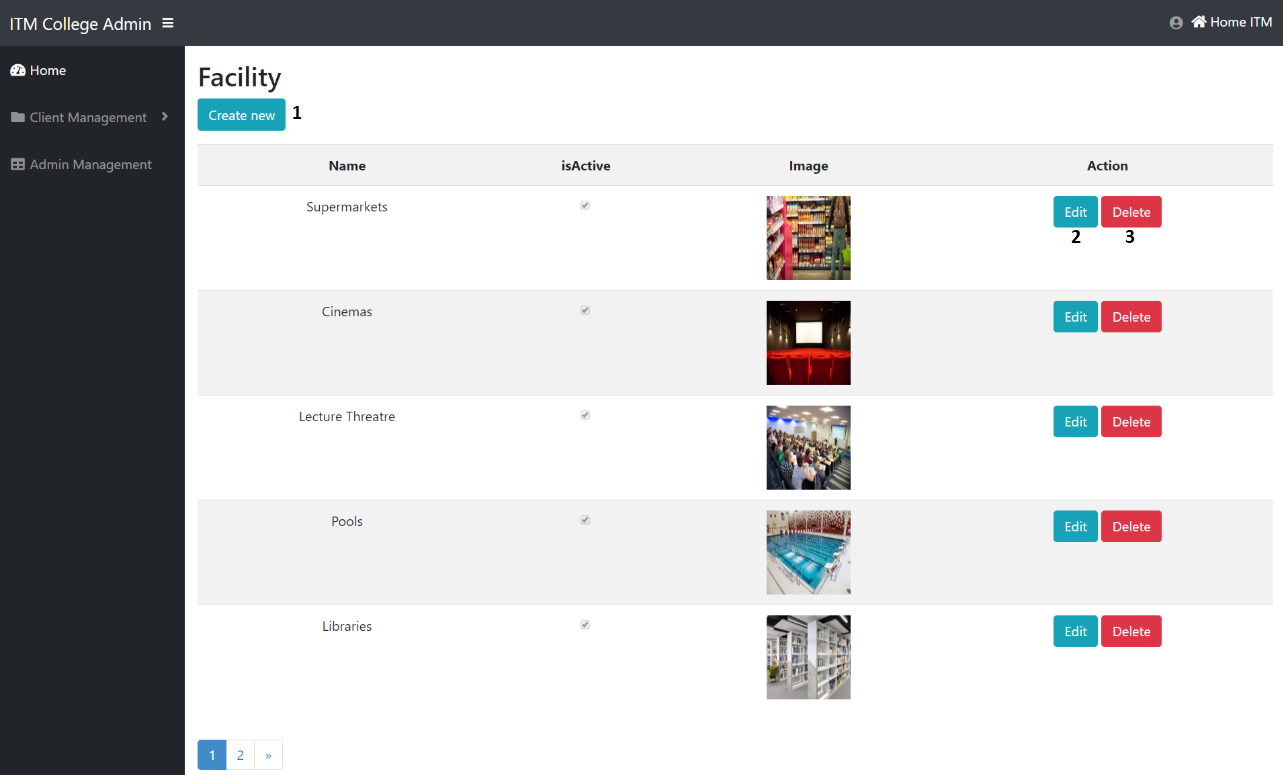
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Edit Faculty Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | Name | TextField | 0 <= Length <= 50 | Input | Input Faculty Name | Enable |
| 2 | DoB | TextField |  | Input | Input Faculty Day of Birth | Enable |
| 3 | Degree | TextField | 0 <= Length <= 20 | Click | Input Degree of Faculty | Enable |
| 4 | Department | Dropdown |  | Click | Choose Department | Enable |
| 5 | Choose File | File | Required | Click | Path of Image | Enable |
| 6 | btnSave | Button |  | Click | Submit form Edit | Enable |
| 7 | Back to list | Hyberlink |  | Click | View faculty list | Enable |

* Some code review:
* Edit Faculty:

****

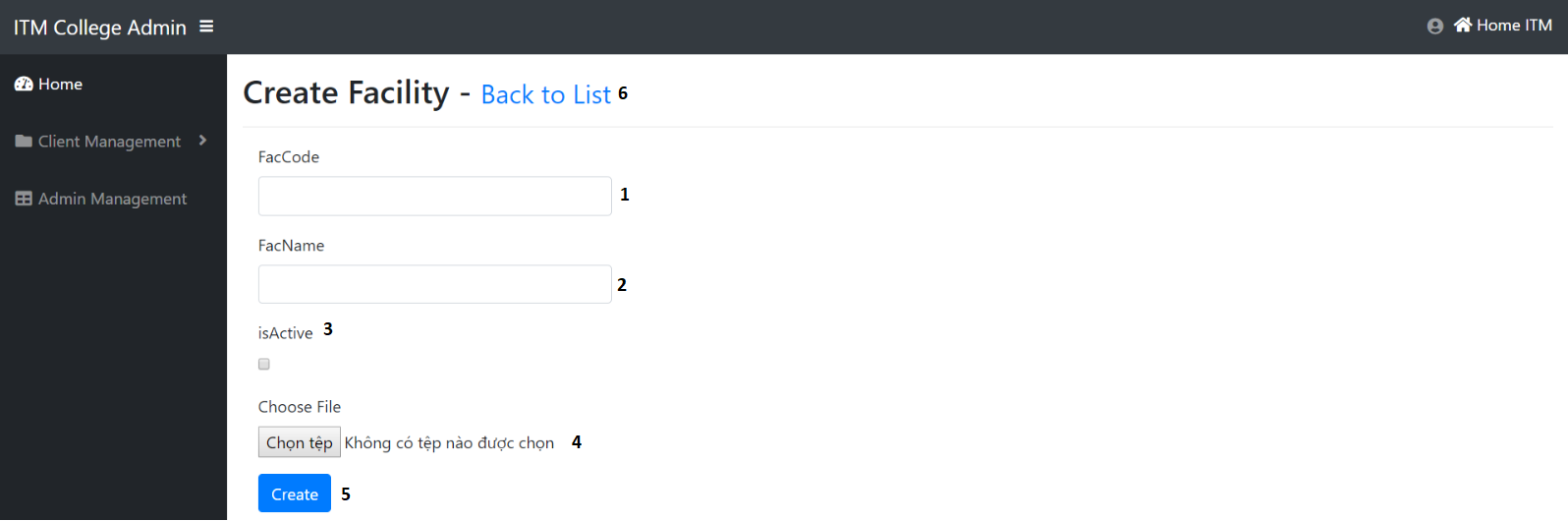
1. Facility management

**a. Facility Post List**

****

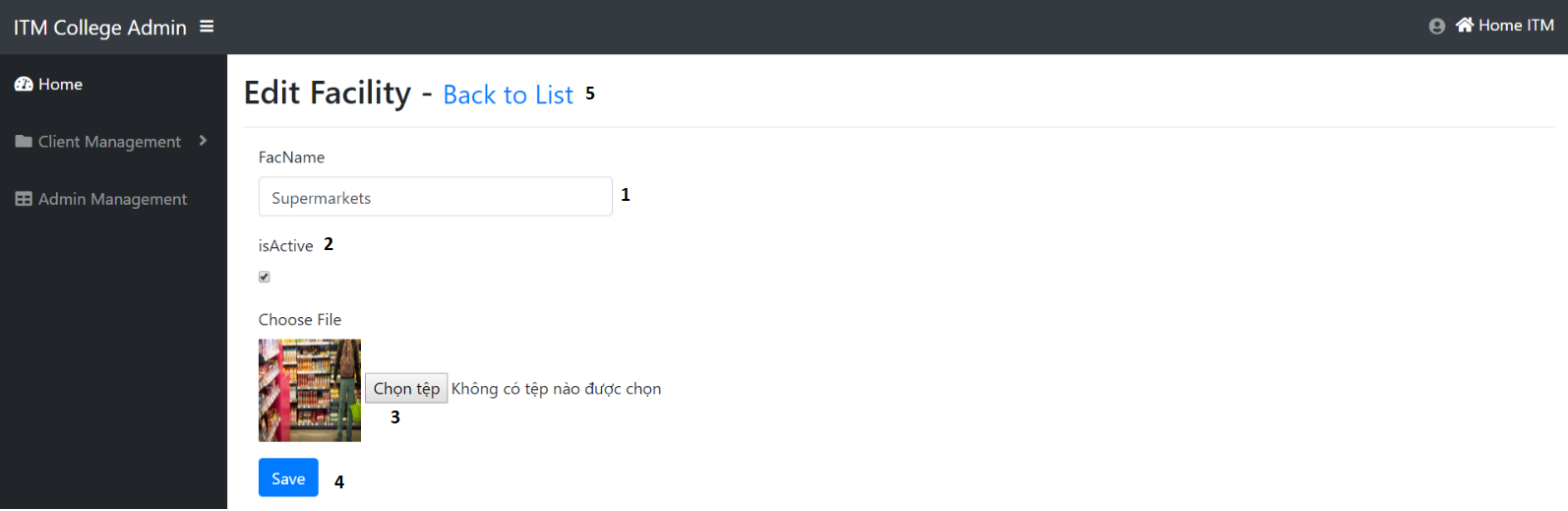
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Facility Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | Create | Hyberlink |  | Click | Create new Facility Post | Enable |
| 3 | btnEdit | Button |  | Click | Edit Facility Post | Enable |
| 4 | btnDelete | Button |  | Click | Delete Facility Post | Enable |

**b. Create Facility Post**

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Create Facility Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | FacCode | TextField | 0 <= Length <= 30 | Input | Input Facility ID | Enable |
| 2 | Name | TextField | 0 <= Length <= 50 | Input | Input Facility Name | Enable |
| 3 | isActive | Checkbox |  | Click | Check if Facility is Active | Enable |
| 4 | Choose File | File | Required | Click | Path of Image | Enable |
| 5 | btnCreate | Button |  | Click | Submit form Create | Enable |
| 6 | Back to list | Hyberlink |  | Click | View facility list | Enable |

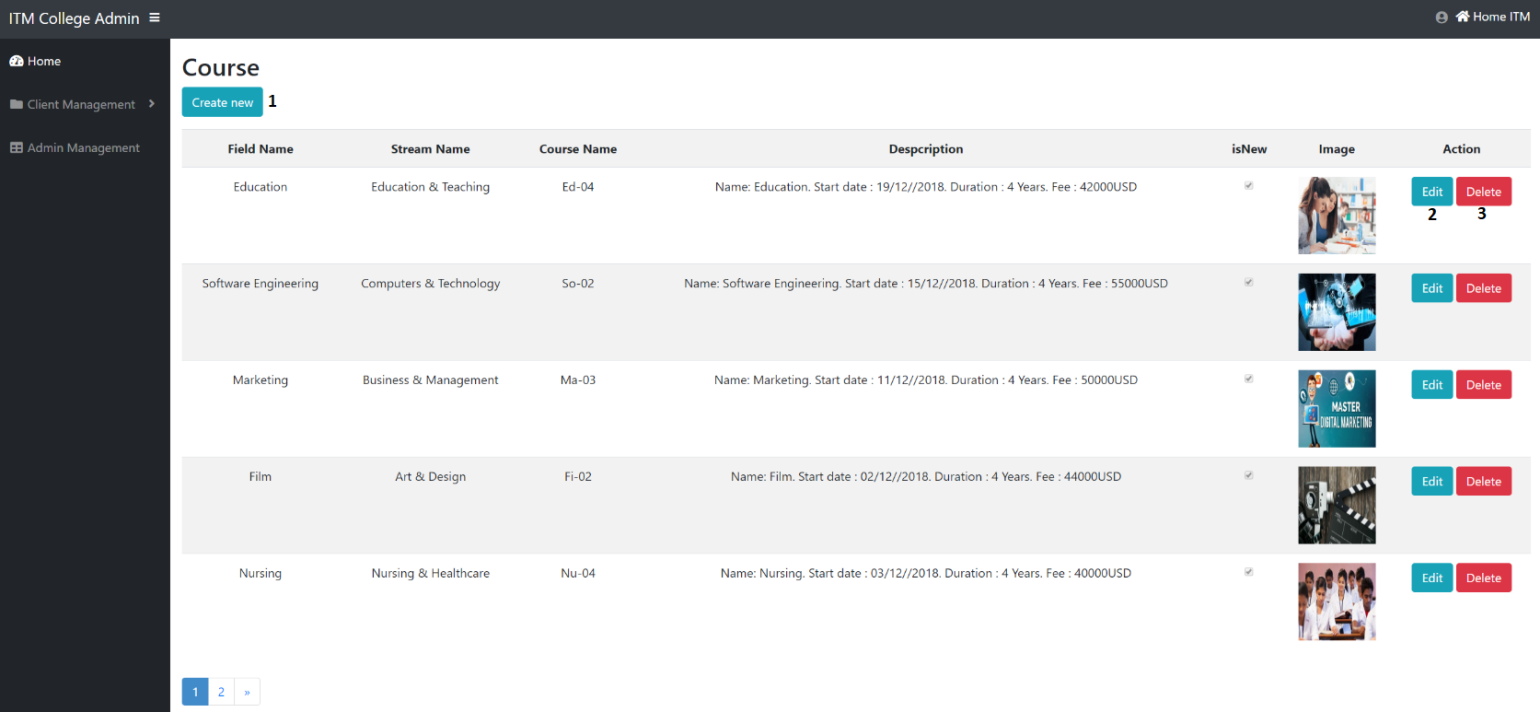
**c. Edit Facility Post**

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Edit Facility Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | Name | TextField | 0 <= Length <= 50 | Input | Input Facility Name | Enable |
| 2 | isActive | RadioButton |  | Click | Check if Facility is Active | Enable |
| 3 | Choose File | File | Required | Click | Path of Image | Enable |
| 4 | btnSave | Button |  | Click | Submit form Edit | Enable |
| 5 | Back to list | Hyberlink |  | Click | View facility list | Enable |

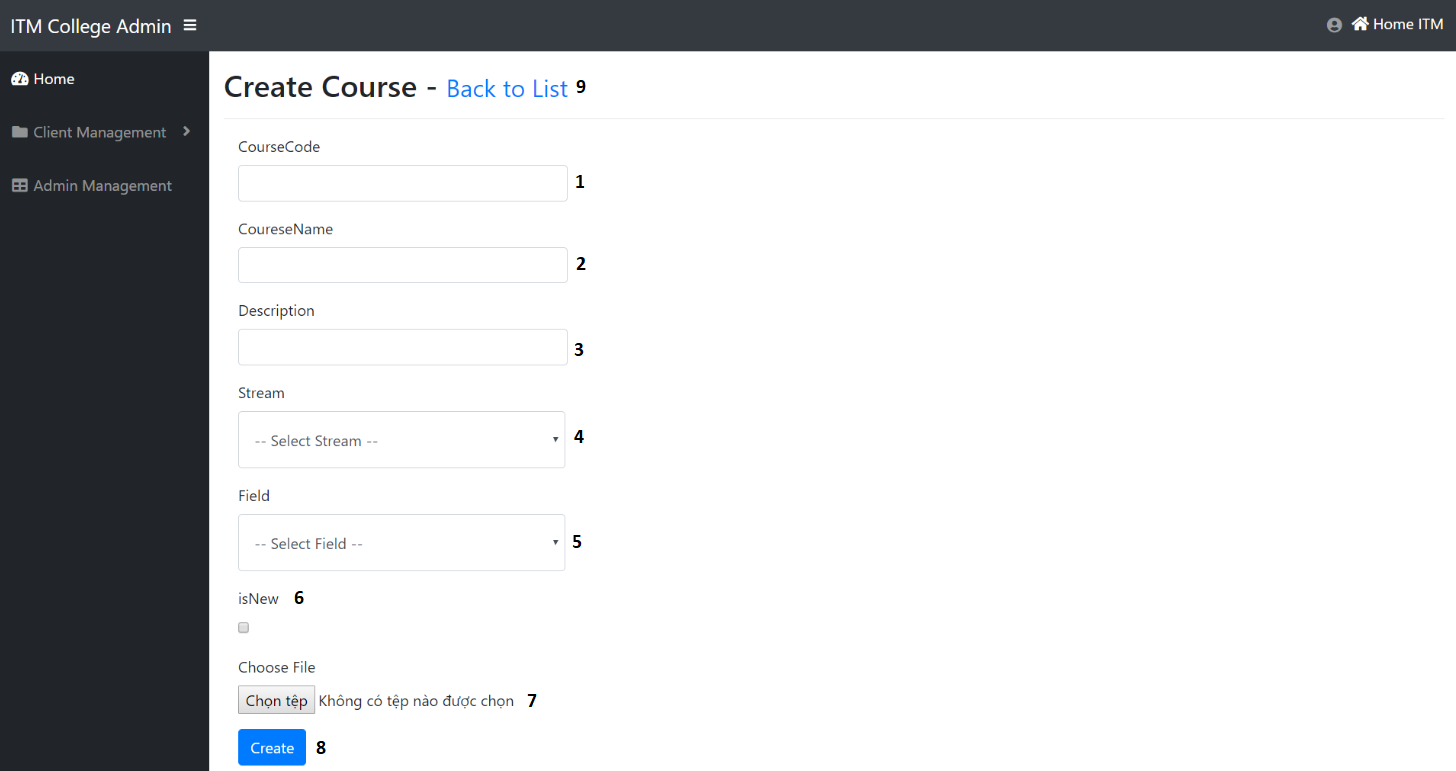
1. Course management

**a. Course Post List**

****

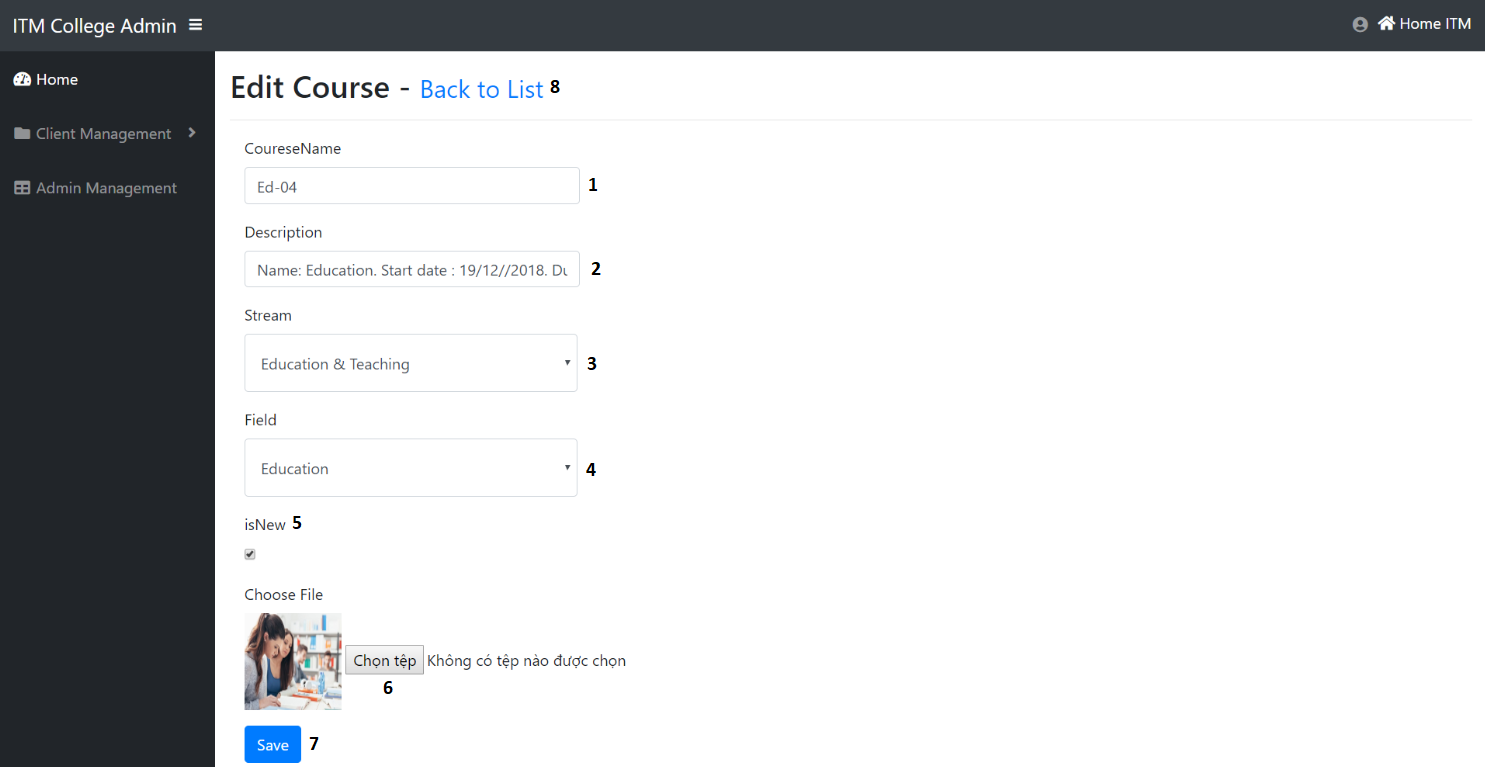
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Course Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | Create | Hyberlink |  | Click | Create new Course Post | Enable |
| 3 | btnEdit | Button |  | Click | Edit Course Post | Enable |
| 4 | btnDelete | Button |  | Click | Delete Course Post | Enable |

**b. Create Course Post**

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Create Course Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | CourseID | TextField | 0 <= Length <= 30 | Input | Input Course ID | Enable |
| 2 | Name | TextField | 0 <= Length <= 50 | Input | Input Course Name | Enable |
| 3 | Description | TextField |  | Input | About Course | Enable |
| 4 | Stream | Dropdown |  | Click | Choose Stream for Course | Enable |
| 5 | Field | Dropdown |  | Click | Choose Field for Course | Enable |
| 6 | isNew | RadioButton |  | Click | Check if Course is New | Enable |
| 7 | Choose File | File | Required | Clcik | Path of Image | Enable |
| 8 | btnCreate | Button |  | Click | Submit form Create | Enable |
| 9 | Back to list | Hyberlink |  | Click | View Course list | Enable |

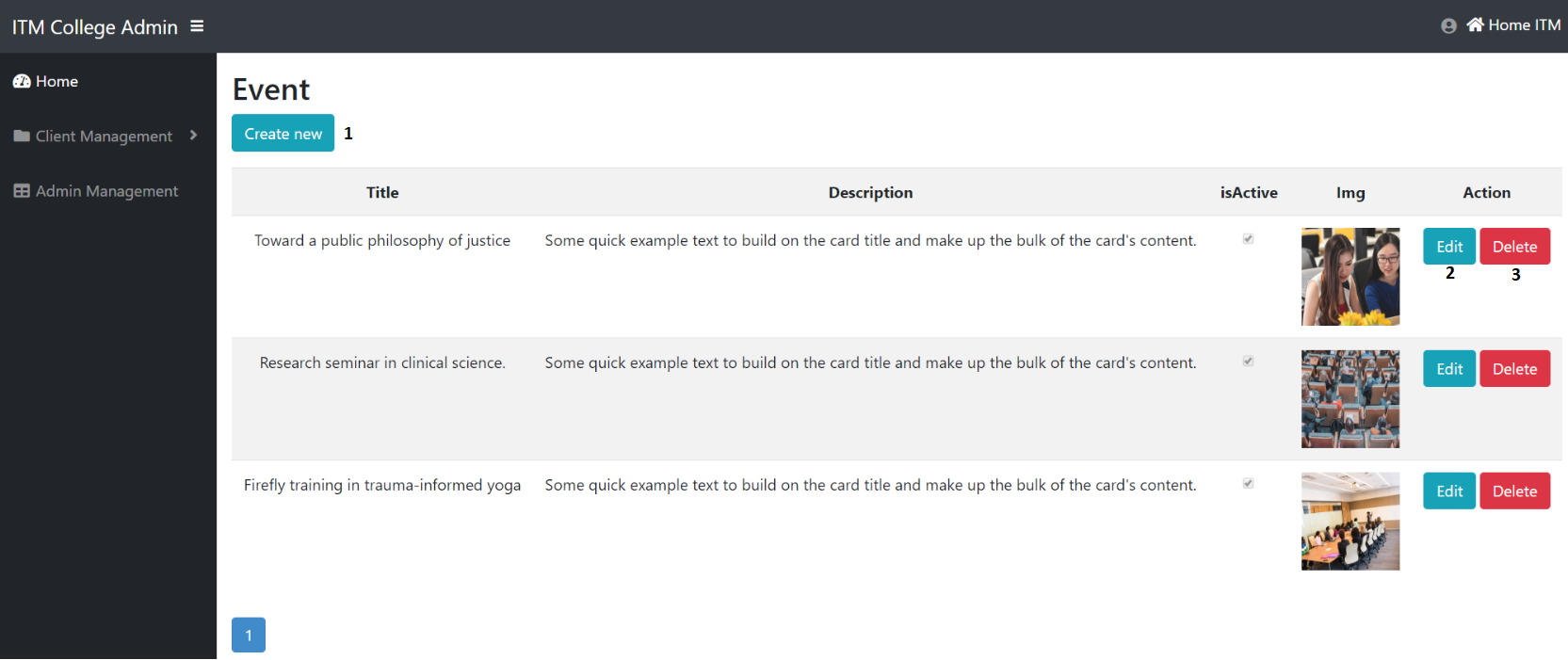
**c. Edit Course Post**

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Edit Course Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | Name | TextField | 0 <= Length <= 50 | Input | Input Course Name | Enable |
| 2 | Description | TextField |  | Input | About Course | Enable |
| 3 | Stream | Dropdown |  | Click | Choose Stream for Course | Enable |
| 4 | Field | Dropdown |  | Click | Choose Field for Course | Enable |
| 5 | isNew | RadioButton |  | Click | Check if Course is New | Enable |
| 6 | Choose File | File | Required | Click | Path of Image | Enable |
| 7 | btnSave | Button |  | Click | Submit form Edit | Enable |
| 8 | Back to list | Hyberlink |  | Click | View course list | Enable |

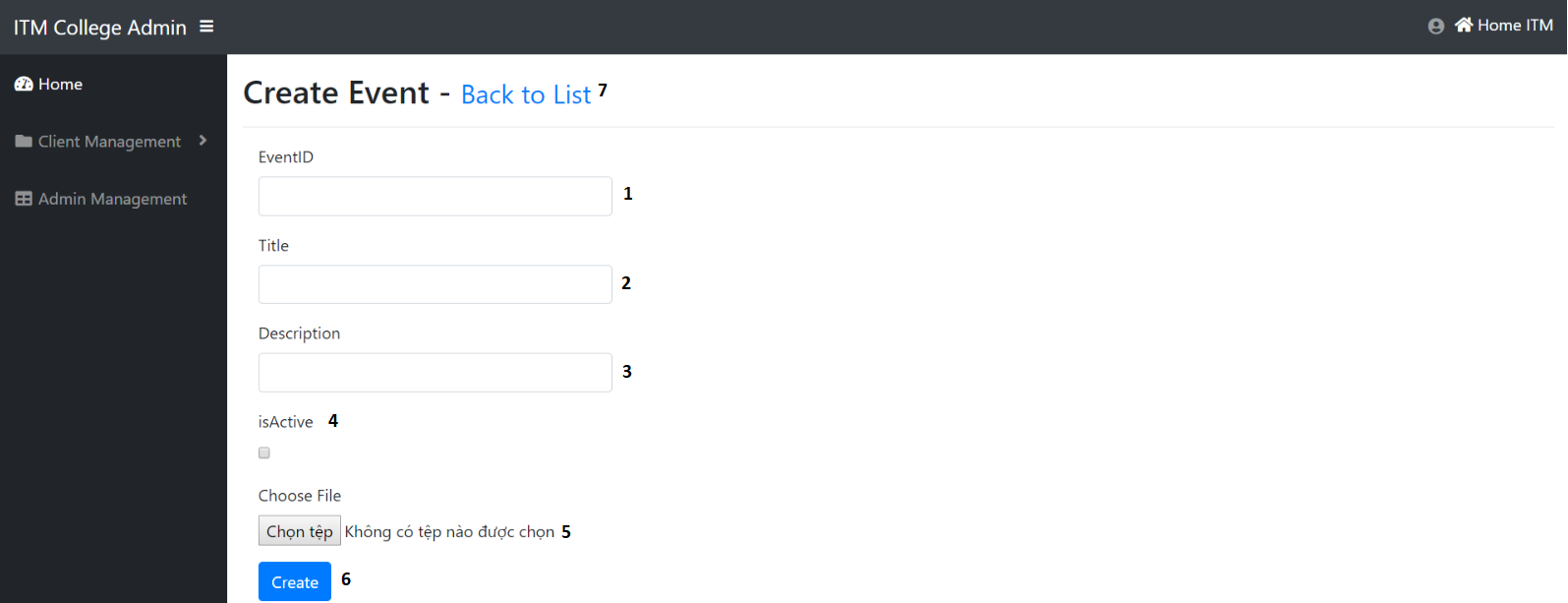
1. Event management

**a. Event Post List**

****

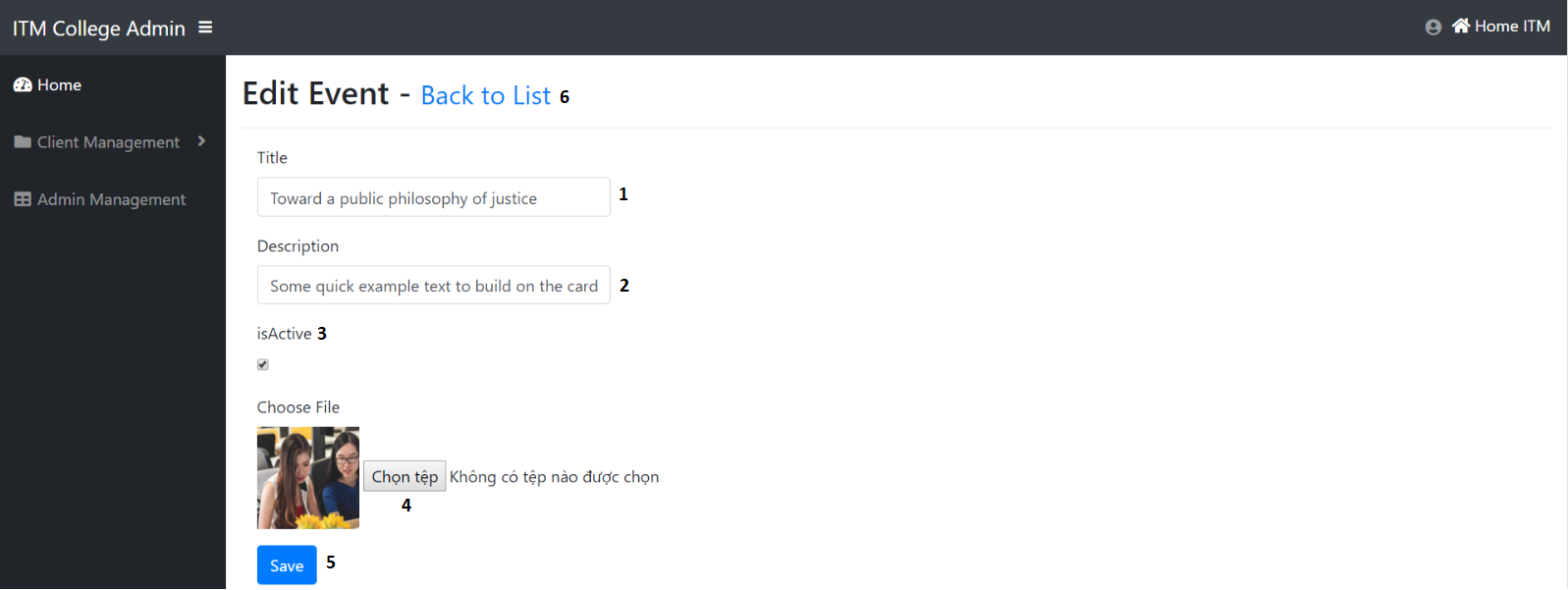
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Event Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | Create | Hyberlink |  | Click | Create new Event Post | Enable |
| 3 | btnEdit | Button |  | Click | Edit Event Post | Enable |
| 4 | btnDelete | Button |  | Click | Delete Event Post | Enable |

**b. Create Event Post**

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Create Event Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | EventID | TextField | 0 <= Length <= 30 | Input | Input Event ID | Enable |
| 2 | Title | TextField | 0 <= Length <= 50 | Input | Input Event Title | Enable |
| 3 | Description | TextField |  | Input | About Event | Enable |
| 4 | isActive | RadioButton |  | Click | Check if Event is Active | Enable |
| 5 | Choose File | File | Required | Click | Path of Image | Enable |
| 6 | btnCreate | Button |  | Click | Submit form Create | Enable |
| 7 | Back to list | Hyberlink |  | Click | View event list | Enable |

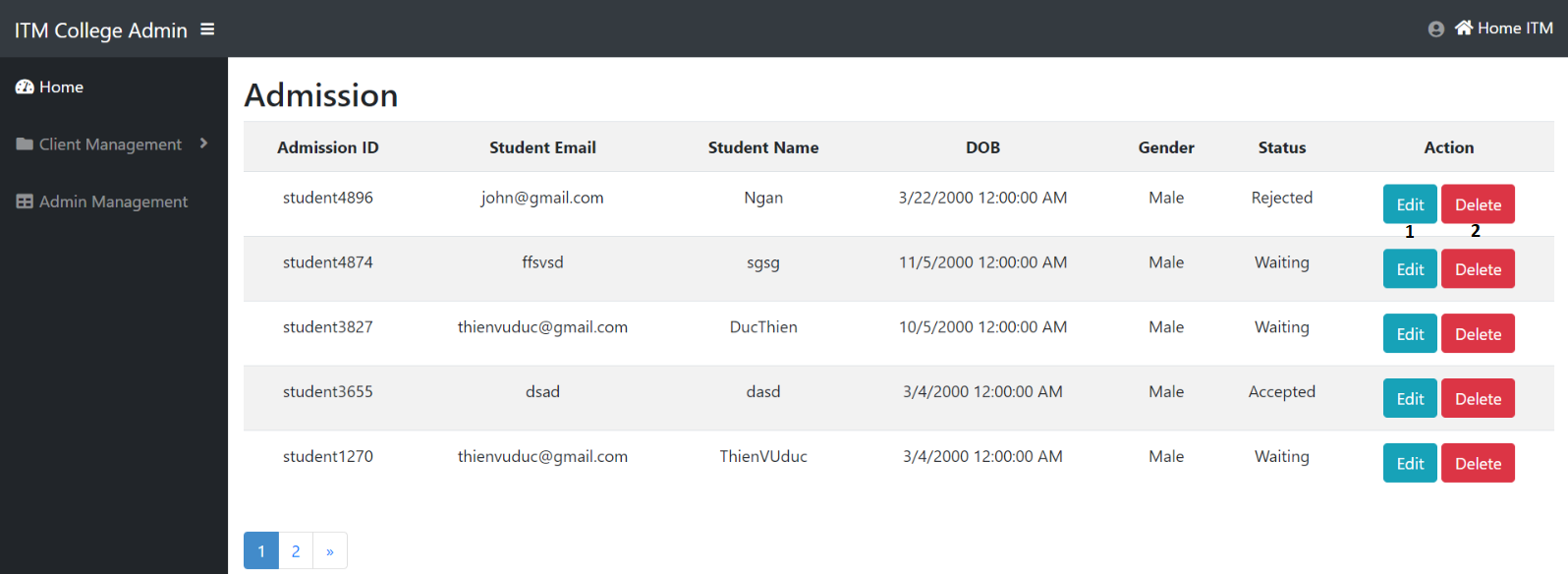
**c. Edit Event Post**

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Edit Event Post | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | Title | TextField | 0 <= Length <= 50 | Input | Input Event Title | Enable |
| 2 | Description | TextField |  | Input | About Event | Enable |
| 3 | isActive | RadioButton |  | Click | Check if Event is Active | Enable |
| 4 | Choose File | File | Required | Click | Path of Image | Enable |
| 5 | btnSave | Button |  | Click | Submit form Edit | Enable |
| 6 | Back to list | Hyberlink |  | Click | View event list | Enable |

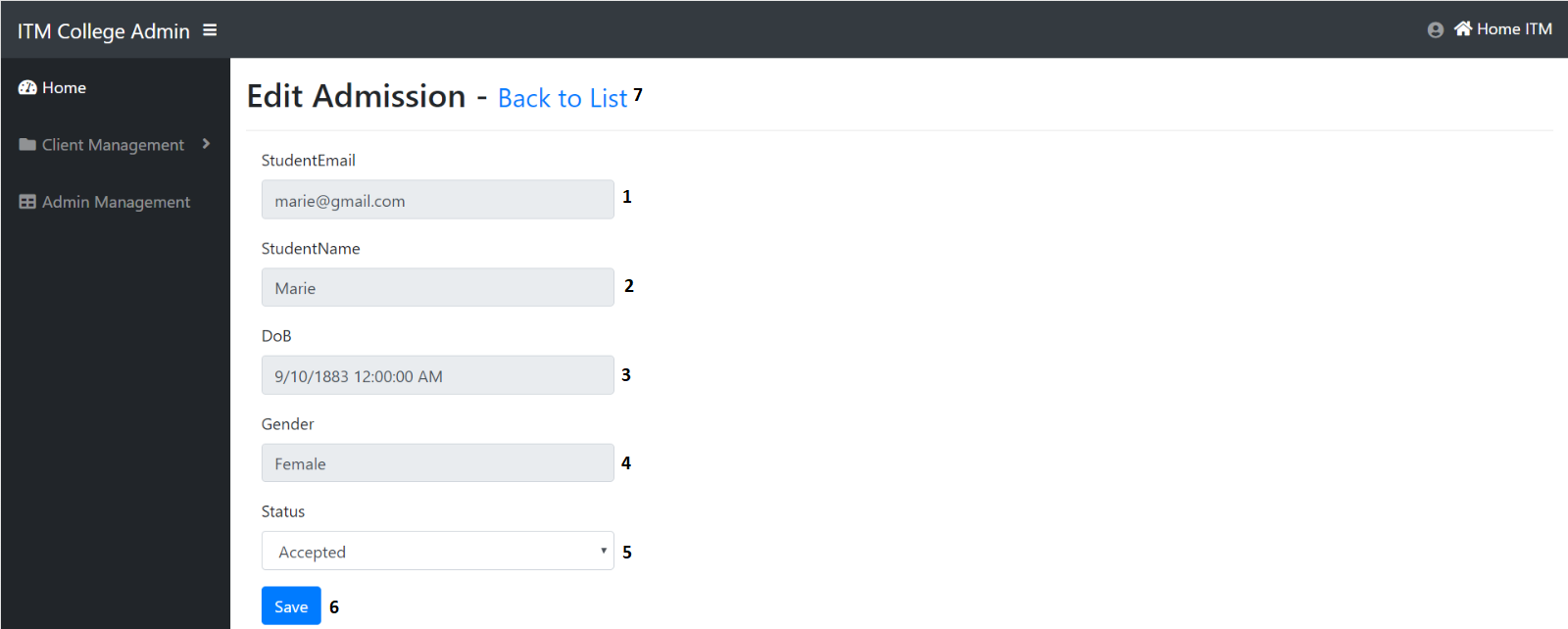
1. Admission management:

a. Admission List



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Admissions List** | | | | | | |
| **No** | Name | Type | Validation | Event | Description | Status |
| **1** | btnEdit | button |  | Click | URL to EditAdmission action | Enable |
| **2** | btnDelete | button |  | Click | Delete Admission | Enable |

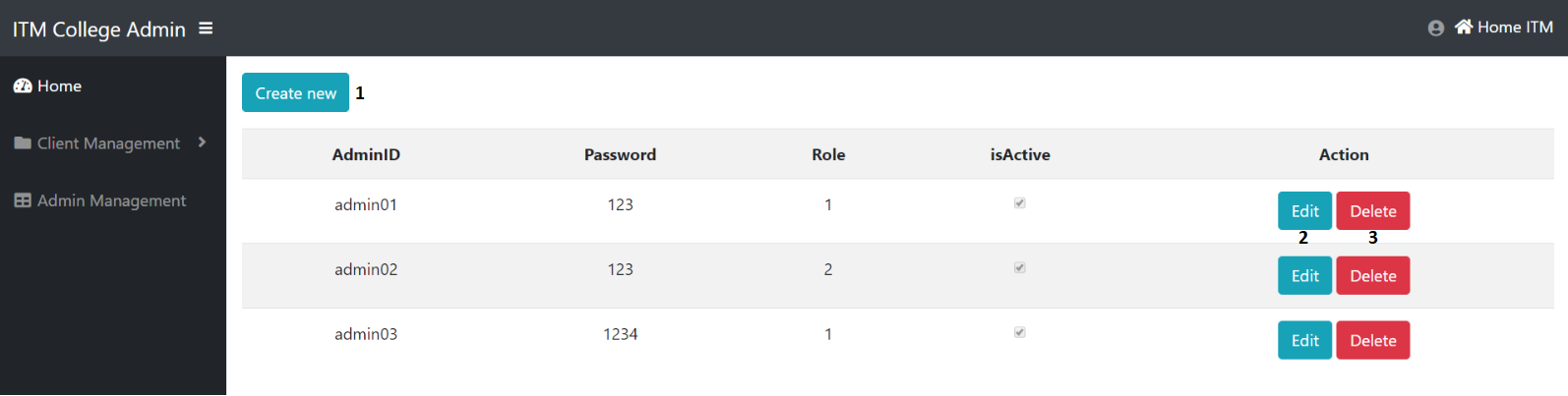
b. Edit Admission Status



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Admission Form** | | | | | | |
| **No** | Name | Type | Validation | Event | Description | Status |
| **1** | StudentEmail | Text box |  |  | Display student email | Readonly |
| **2** | StudentName | Text box |  |  | Display student name | Readonly |
| **3** | DoB | Text box |  |  | Display date of birth | Readonly |
| **4** | Gender | Text |  |  | Display gender | Readonly |
| **5** | Status | Dropdown list |  | Select option | Input status (default = 0 : Waiting,  -1: Rejected,  1: Accepted) | Enable |
| **6** | btnSave | Button |  | Click | Save admission | Enable |
| **7** | Back to list | Hyberlink |  | Click | View Admission list | Enable |

1. Admin management:

a. Admin List



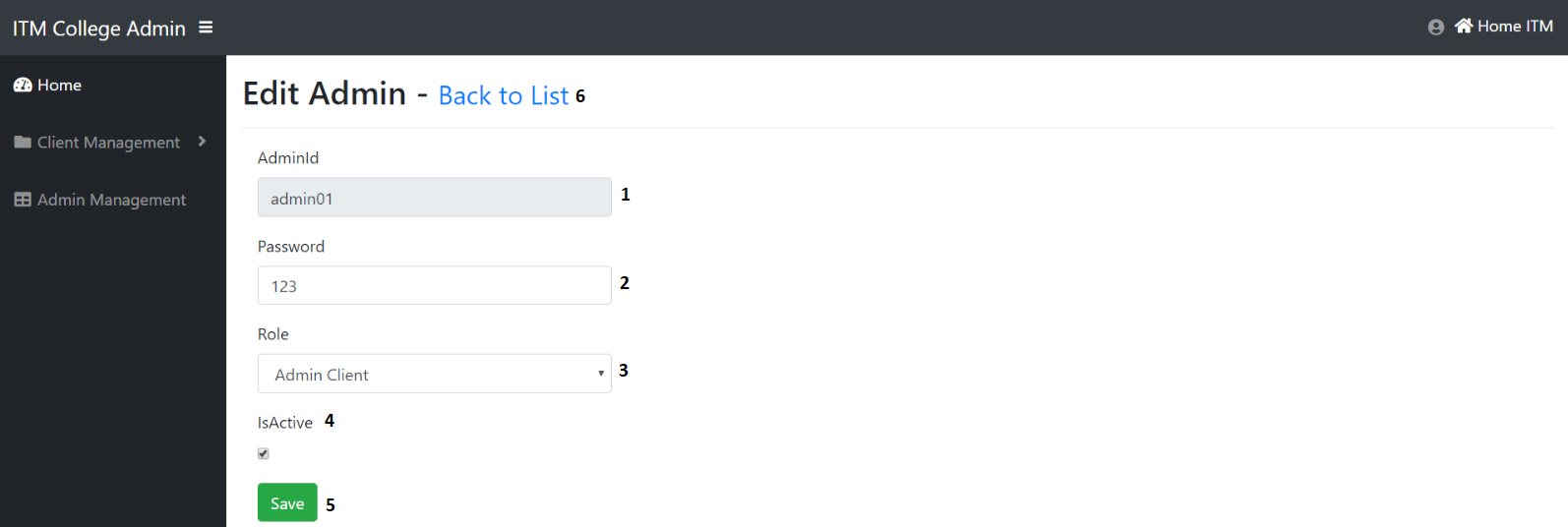
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Admins page** | | | | | | |
| **No** | Name | Type | Validation | Event | Description | Status |
| **1** | btnCreate | Hyperlink |  | Click | Create new admin account | Enable |
| **2** | btnEdit | Button |  | Click | URL to edit action | Enable |
| **3** | btnDelete | Button |  | Click | Delete Admin | Enable |

b. Create Admin Account



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Create admin form** | | | | | | |
| **No** | Name | Type | Validation | Event | Description | Status |
| **1** | AdminID | Text box | Not null | Input | Input admin ID | Enable |
| **2** | Password | Text box | Not null | Input | Input admin password | Enable |
| **3** | Role | Dropdown list |  | Input | Input admin role | Enable |
| **4** | isActive | Check box |  | Click | Input account status | Enable |
| **5** | btnCreate | Button |  | Click | Submit form | Enable |
| **6** | Back to list | Button |  | Click | Url to Index action (Areas/Admin/User controller ) | Enable |

c. Edit Admin Account

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Edit admin form** | | | | | | |
| **No** | Name | Type | Validation | Event | Description | Status |
| **1** | AdminID | Text box |  |  | Display Admin ID | Disable |
| **2** | Password | Text box | Not null | Input | Input admin password | Enable |
| **3** | Role | Dropdown list |  | Input | Input admin role | Enable |
| **4** | isActive | Check box |  | Click | Input account status | Enable |
| **5** | btnSave | Button |  | Click | Submit form | Enable |
| **6** | Back to list | Button |  | Click | Url to Index action (Areas/Admin/User controller ) | Enable |

**Task Sheet Review III**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | Database Diagram | Jul-25,2022 | 2 | Nguyen Hoang Thien An | Completed |
| 02 | Database Structure | Vo Phan Hien | Completed |
| 03 | GUI Design Front-End | Vo Phan Hien | Completed |

|  |  |  |
| --- | --- | --- |
|  | Prepare By: Group 1 | Approved By: Faculty |
| Date: Aug,1st,2022 | Team Leader  Nguyen Hoang Thien An | Mr. Hoang Duc Quang |