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

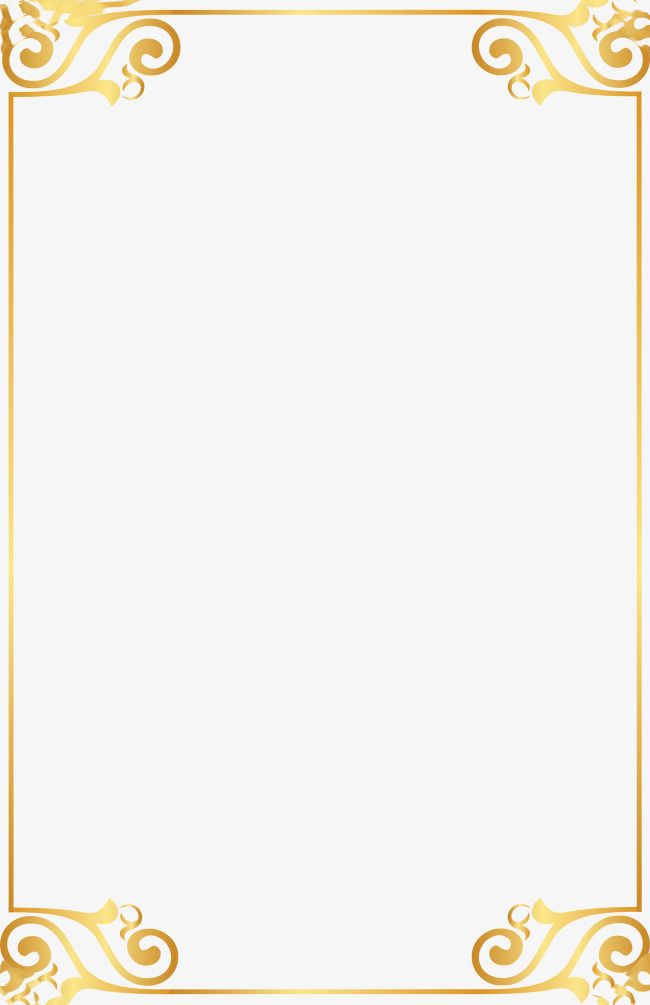
**Centre Name: ACE-HCMC-2-FPT.**

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

**Online Help Desk  
(OHD)**

|  |  |  |
| --- | --- | --- |
| **Supervisor:** | MR. NGO PHUOC NGUYEN | |
| **Semester:** | 3 | |
| **Batch No:** | T1.1908.E0 | |
| **Group No:** | 4 | |
| **Order:** | **Full name** | **Roll No.** |
| 1. | Nguyen Minh Thoai | Student1225526 |
| 2. | Duong Ton Truc | Student1225628 |
| 3. | Truong Ngo Chi | Student1200855 |
| 4. | Hoang Thien An | Student1198581 |

December 2020



This is to certify that

Mr. Nguyen Minh Thoai

Mr. Duong Ton Truc

Mr. Truong Ngo Chi

Mr. Hoang Thien An

have successfully designed and developed

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

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

Submitted by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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**

1. **Introduction**

This project is aimed at developing an Online Help Desk (OHD) for the facilities in the campus. This is an Intranet based application that can be accessed throughout the campus. This system can be used to automate the workflow of service requests for the various facilities in the campus. This is one integrated system that covers different kinds of facilities like class-rooms, labs, hostels, mess, canteen, gymnasium, computer centre, faculty club etc. Registered users (students, faculty, lab-assistants and others) will be able to log in a request for service for any of the supported facilities. These requests will be sent to the concerned people, who are also valid users of the system, to get them resolved. There are features like email notifications/reminders, addition of a new facility to the system, report generators etc in this system.

.

1. **Project Requirement**

There are registered people in the system (students, faculty, lab-assistants and others). Some of them are responsible for maintaining the facilities (like, the lab-assistant is responsible for keeping the lab ready with all the equipment in proper condition, the students council is responsible for taking forward students’ complaints/requests to the faculty/administration etc).   
There are three kinds of users for this system:

1. those who use the system to create a request (end-users)
2. those who look at the created requests and assign them to the concerned people (facility-heads)
3. those who work on the assigned requests and update the status of the same on the system (assignees)
4. There is also an ‘Administrator’ for doing the Admin-level functions such as creating user accounts, adding new facilities to the system etc.

.

1. **Project Guide**

❖ Decide on the list of facilities that would be supported and define it formally.

❖ Make a database of different kinds of users (End-users, Facility-heads, Assignees).

❖ Create the front-page of the OHD system giving a brief description about the system and a login box.

❖ Create the help-pages of the system in the form of Q&A. This will help you also when implementing the system.

❖ Create other sub-systems like automatic notification, screens for various functions (like create\_new\_request,view\_open\_requests, forward\_new\_request\_to\_assignee etc).

❖ A database to store the requests should be designed. Request Id (which is created automatically by the system, as soon as a request is created) can be a key for this database. The range of valid values entered below as examples need not be taken as such. They can be modified by the team.

❖ Another database to store the user information (such as the user name, login/password of the user, kind of user etc) should be designed as well. Following example is for the Requests database.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Range of valid values for the field** | **Remarks** |
| 1 | Request Id | Any valid 5 digit number (say, 00001 to 99999) | This is the key field of the database. This will be able to get all the details associated with the request from the database. |
| 2 | Requestor | Up to 15 characters in length. | This is the person who created the request. |
| 3 | Facility | Pre-defined set (like library, lab, hostel, mess etc) | It should be possible to include more facilities to the system, in future. |
| 4 | Request Date/Time | Any date/time format | This field will give the details about the date/time when a request was created. |
| 5 | Assignee | Up to 15 characters in length | This is the person who would be working on the request. This person is chosen by the respective facility-heads. |
| 6 | Status | Pre-defined set (like open, closed, assigned, work-in-progress, need-more-info, rejected etc) | This field is updated by several people (facility-head updates it to ‘assigned’ once he assigns it to somebody, the assignee updates it to work-in-progress or need-more-info (in case he needs more info to work on the request) or rejected (in case, it is an invalid request). |
| 6 | Remarks | Up to 100 characters in length | Short remark on the request. Can be updated by the End-user/Assignee/ facility-heads. This field can be increased in size, if needed. |

Note: student can add more tables as per the requirement.

1. **Functional Logic**

❖ A person should be able to :

- Login to the system through the first page of the application.

- Change the password after logging into the system.

- See the status of the requests created by him/her (the status could be one of unassigned/assigned/work in progress/closed/rejected).

- See the list of requests (both open and closed) created by him/her over the past.

- Create a new request by specifying the facility, the severity of the request (there may be several levels of severity defined) and a brief description of the request.

- Close a request created by him/her by giving an appropriate reason.

- See the requests that are assigned to him/her by the facility-heads and update the status of requests (after working on them).

- View the incoming requests (if he/she is a facility-head) and assign them to registered users of the system.

- Get help about the OHD system on how to use the different features of the system.

❖ As soon as a request is created, a message will be displayed to the person who created the request and the concerned facility-head.

❖ Similarly, when any status-change occurs for a request (such as the request getting completed etc), an automatic message will be updated to the person who created the request and the concerned facility-head.

❖A summary report on the requests that came in and requests that were serviced should be sent to every facility-head periodically (say, once in a month).**Hardware / Software Requirements**

1. **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 2017 or higher |
| **ASP.Net MVC** | MVC 5 |
| **IIS server** |  |
| **Notepad/Java editor** |  |
| **Python** | Latest version |
| **Javascript** |  |
| **j2sdk1.4.1\_02 (or later).** |  |

1. **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) |

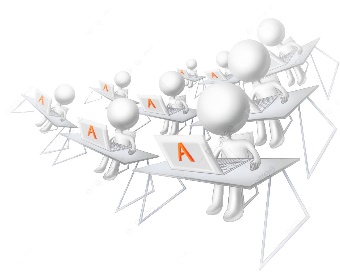
**Task Sheet Review I**

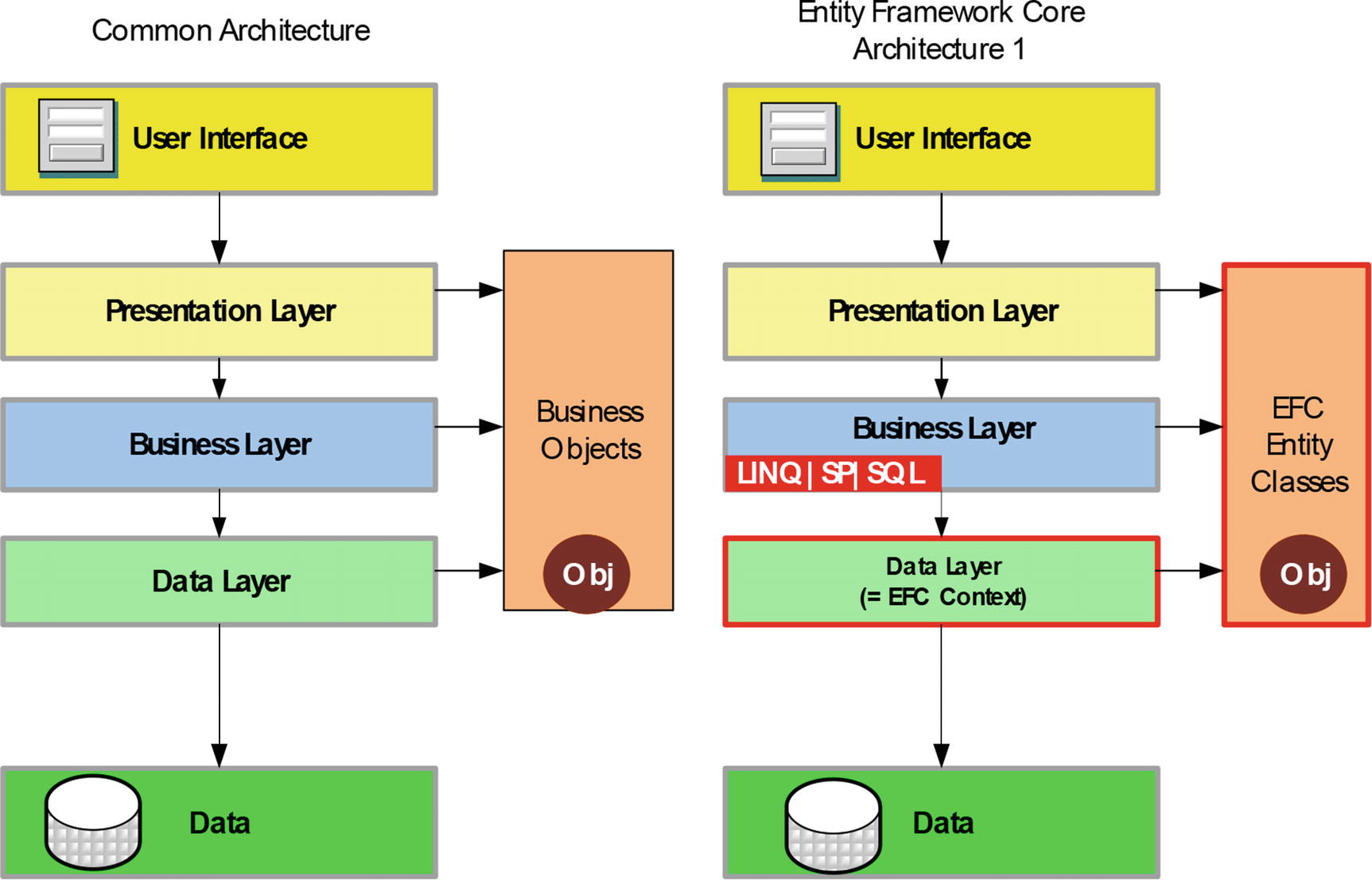
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Ref. No: 4 | Project Title:  Railway Reservation Manage System | Date of Preparation of Activity Plan | | | |
| No. | Task | Actual Start Date | Actual Days | Team Member Names | Status |
| 01 | Acknowledgment | Dec 28, 2020 | 5 | All 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 4 | Approved By: Faculty |
| Date: Dec 28, 2020 | Team Leader  Nguyen Minh Thoai | Mr. Ngo Phuoc Nguyen |

**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: ASP.NET MVC5, Razor, 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.

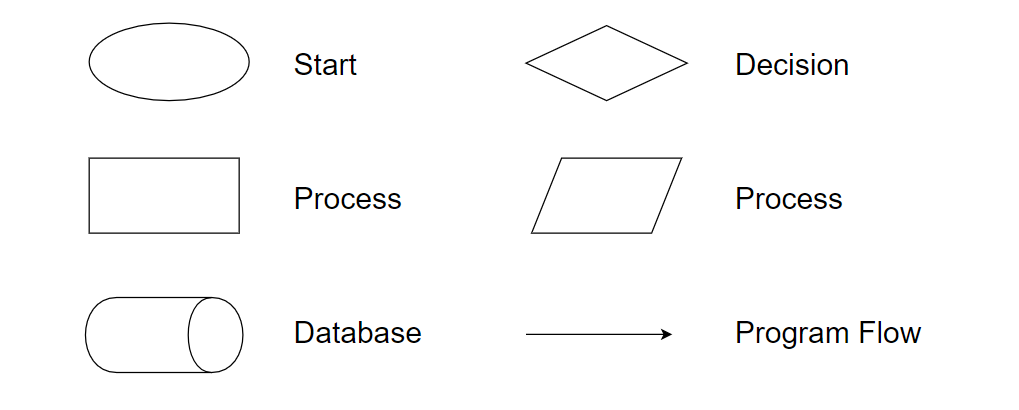
Technology: ASP.NET

## Data Access Tier:

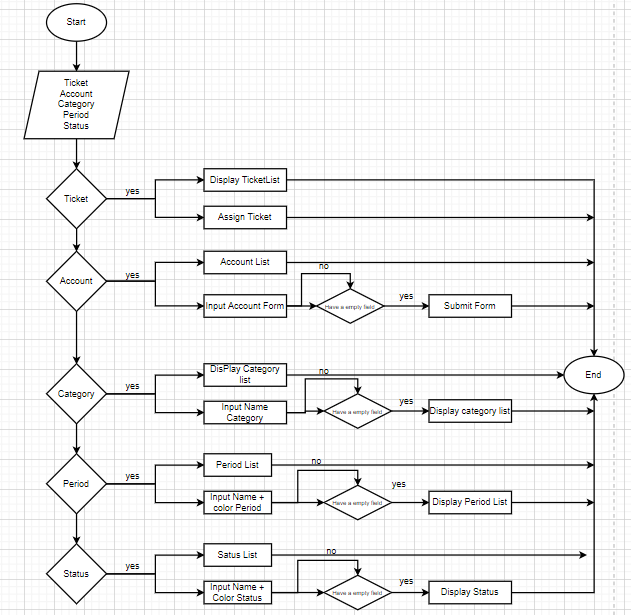
Is basically the server which stores all the application’s data. Data tier contents Database Tables, Database Views and other means of storing Application Data .

Technology: SQL Server, LINQ,..**Algorithms - Data Flowchart**

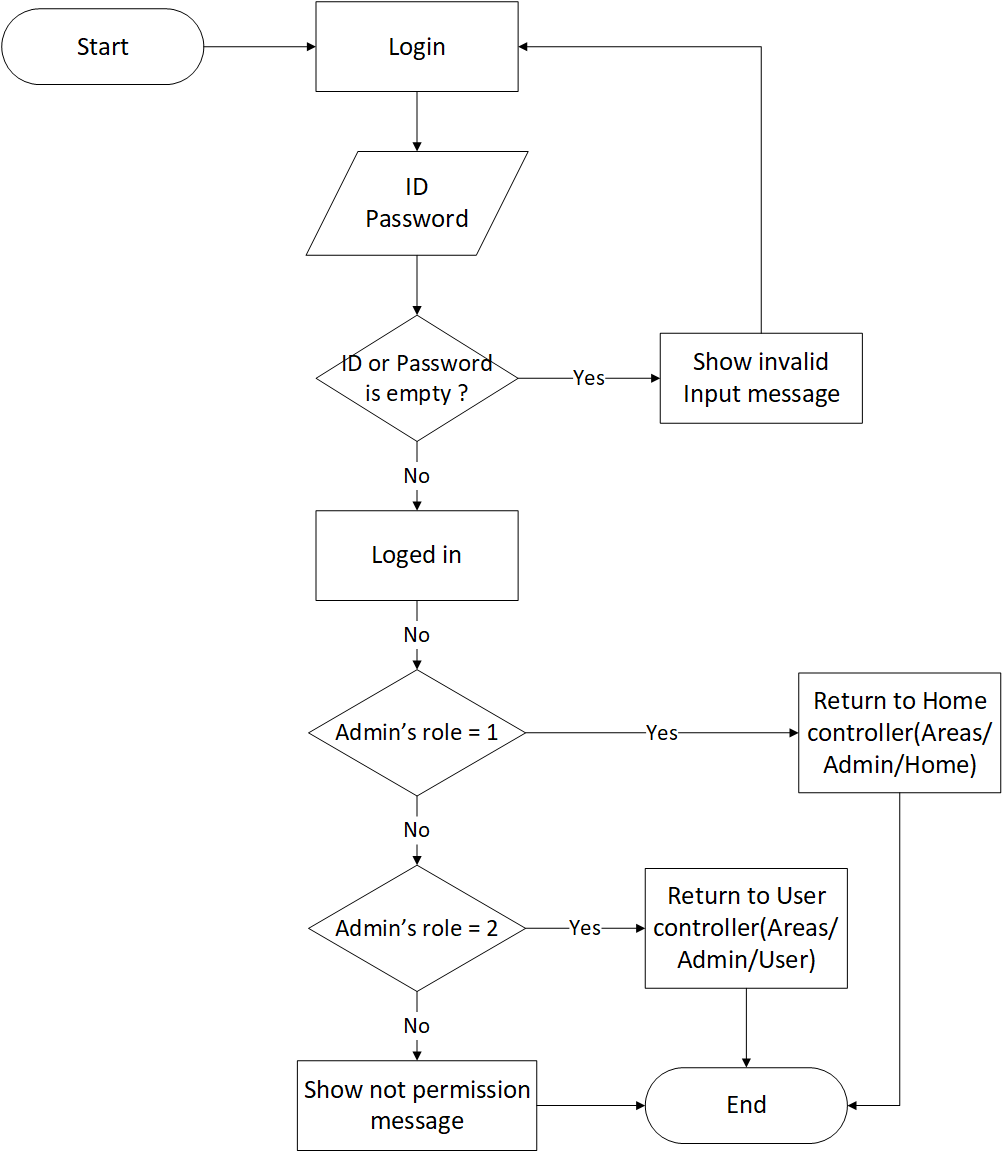
Symbol generates:

****

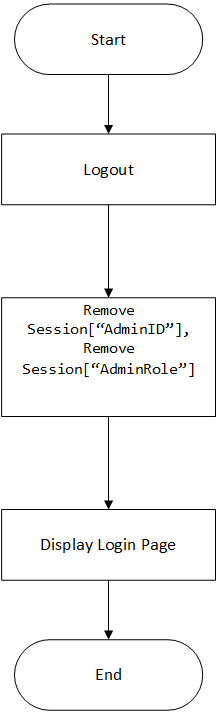
## Main data flow for guest



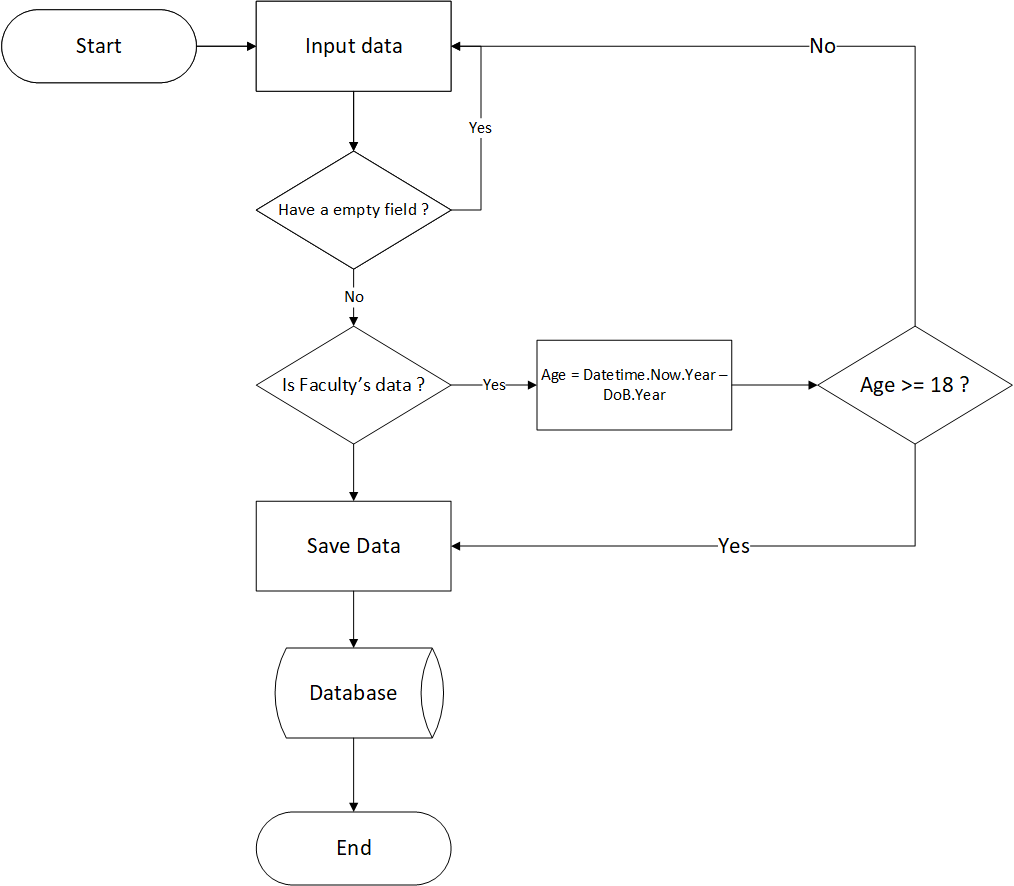
## Login Function (Admin only)



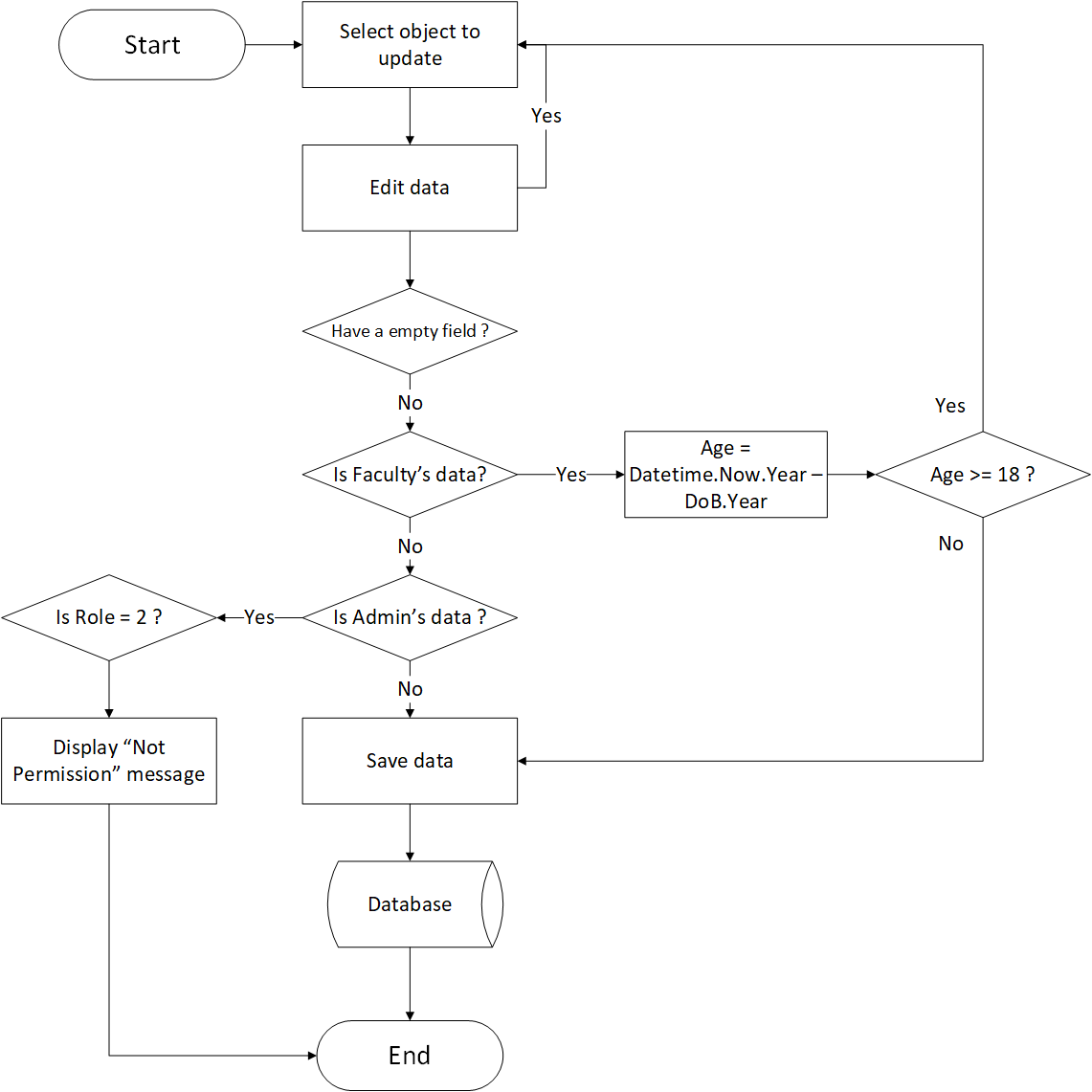
## Logout Function (Admin only)



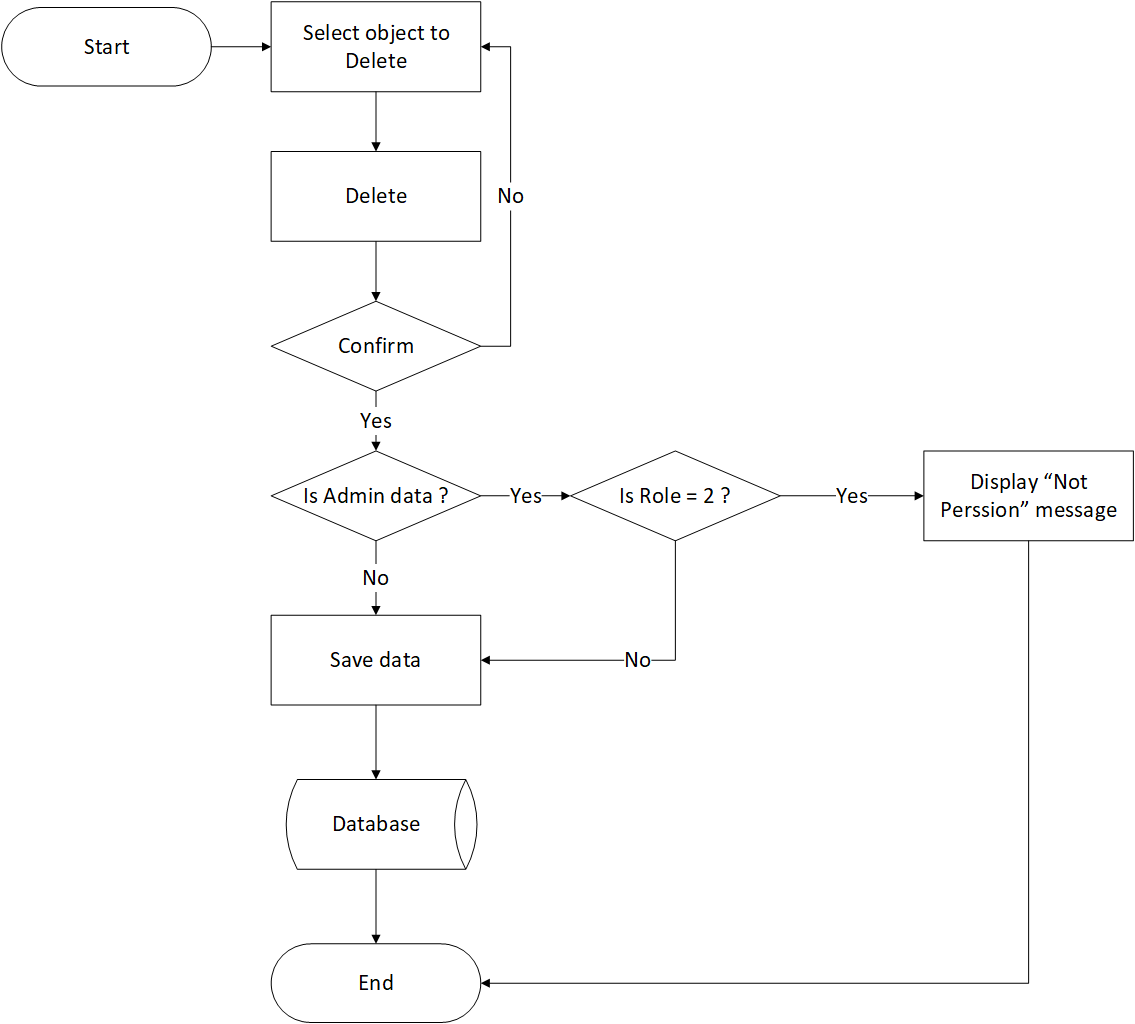
## Insert Data Function (Admin only)



## Update Data Function (Admin only)



## Delete Data Function (Admin only)

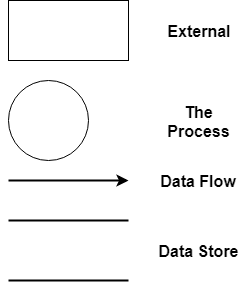


1. **Data Flow Diagram**

**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:**



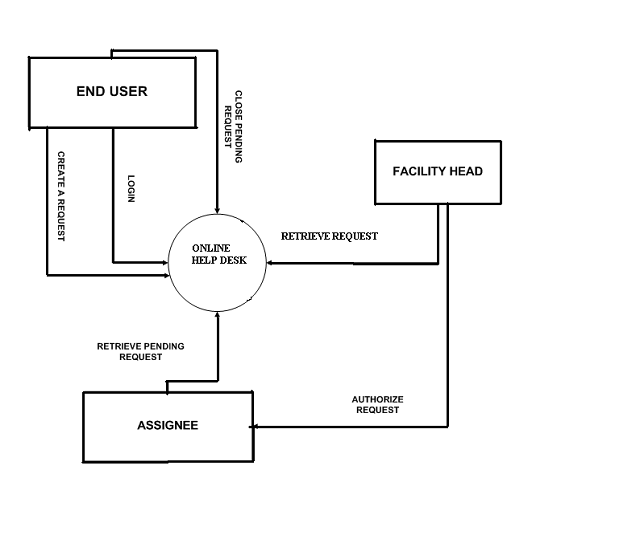
The external factors: External factors can be a person, a group of persons or an organization that are sources of information for the systems and are where system products are transferred to.

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:

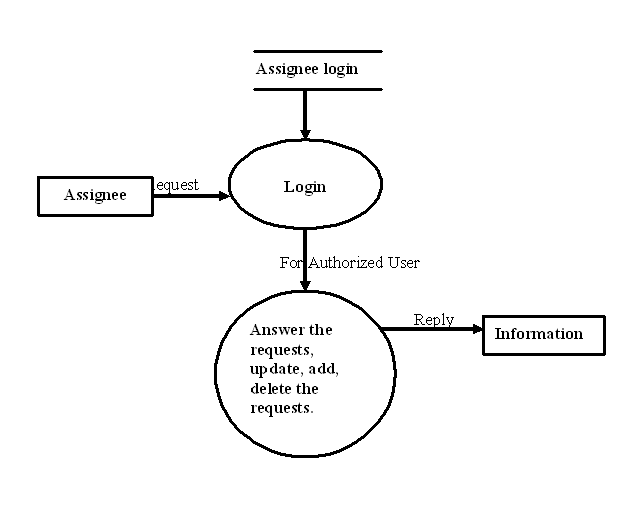


## Level 0 DFD

## User Login

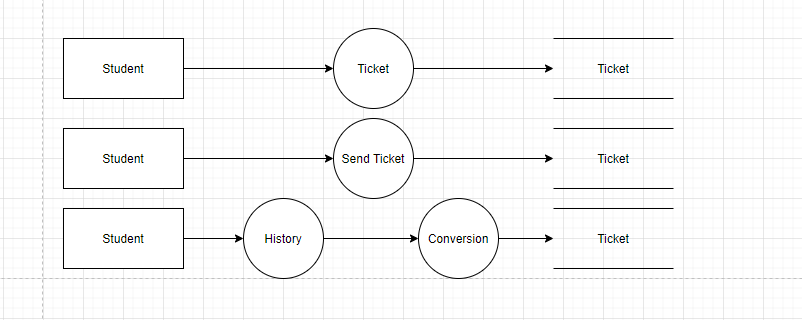
## 0TH level.bmp

## Assignee Login

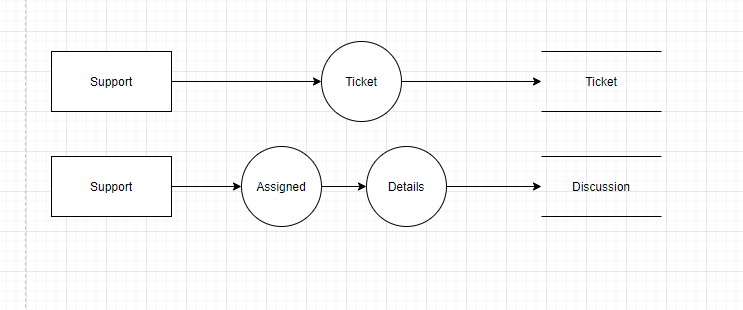


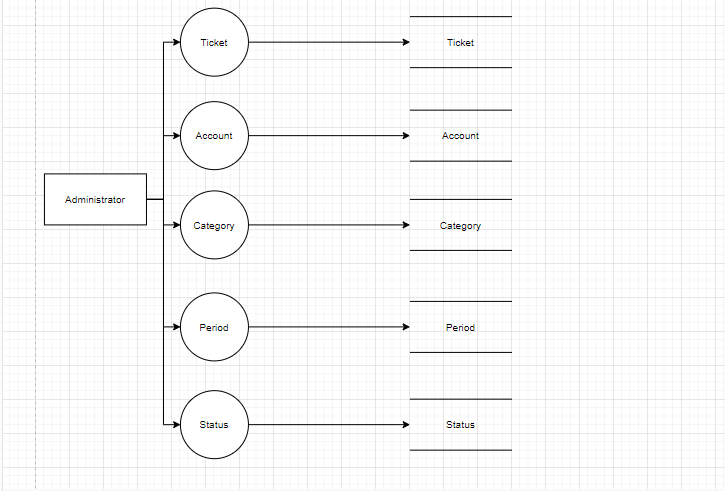
## Level 1 DFD

**3.1 Student Activity:**

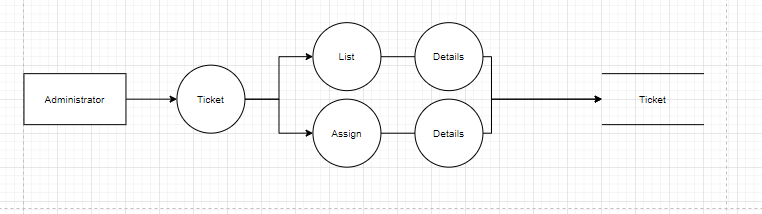


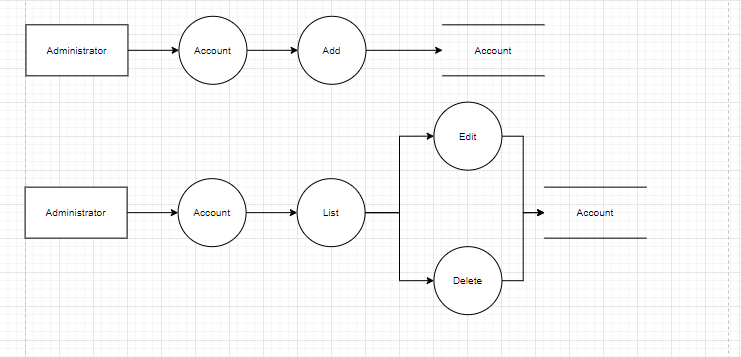
**3.2 Support Activity**



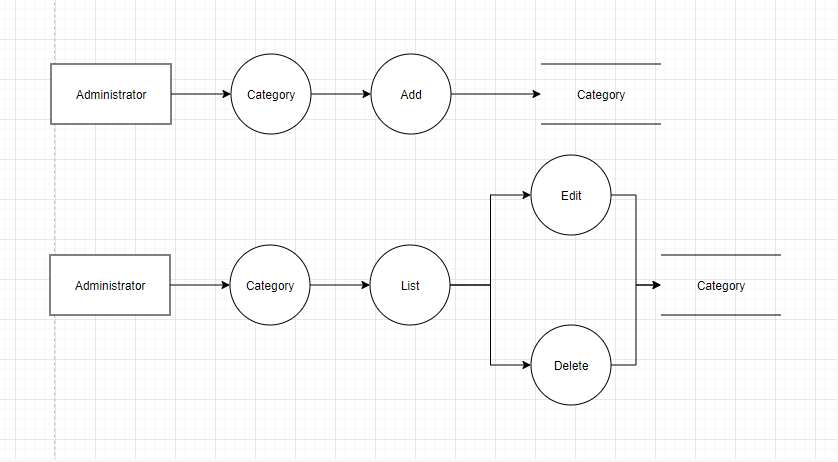
 **3.3 Admin Activity**

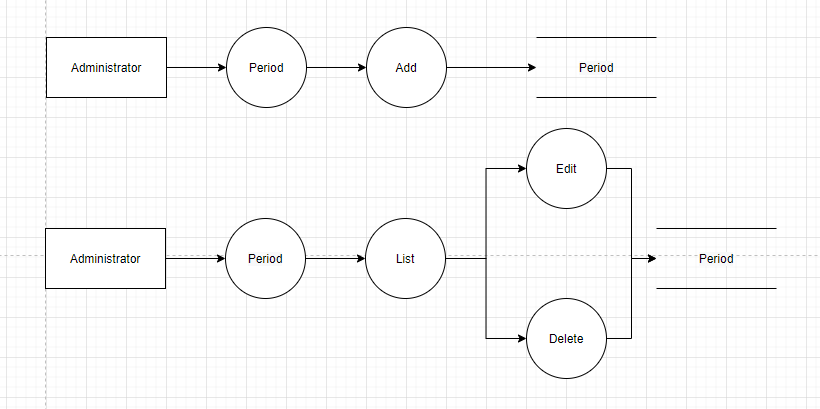
**3.4 Ticket Management**



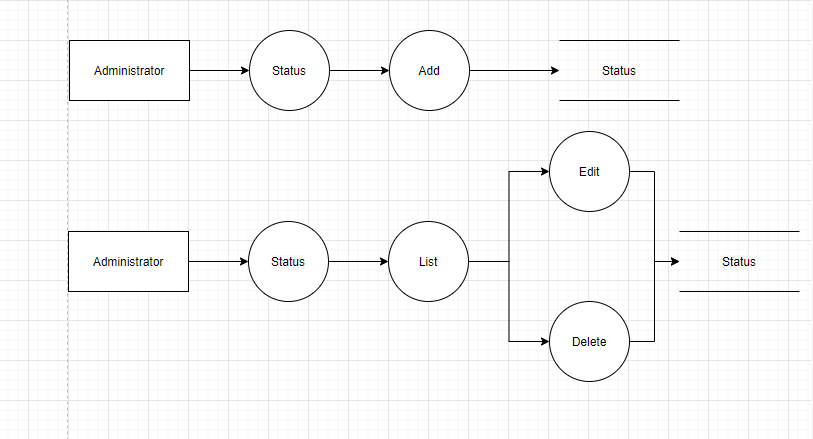
 **3.5 Account Management**

**3.6 Category Management**



 **3.7 Period Management**

**3.7 Status Management**



**Use Case Diagram**

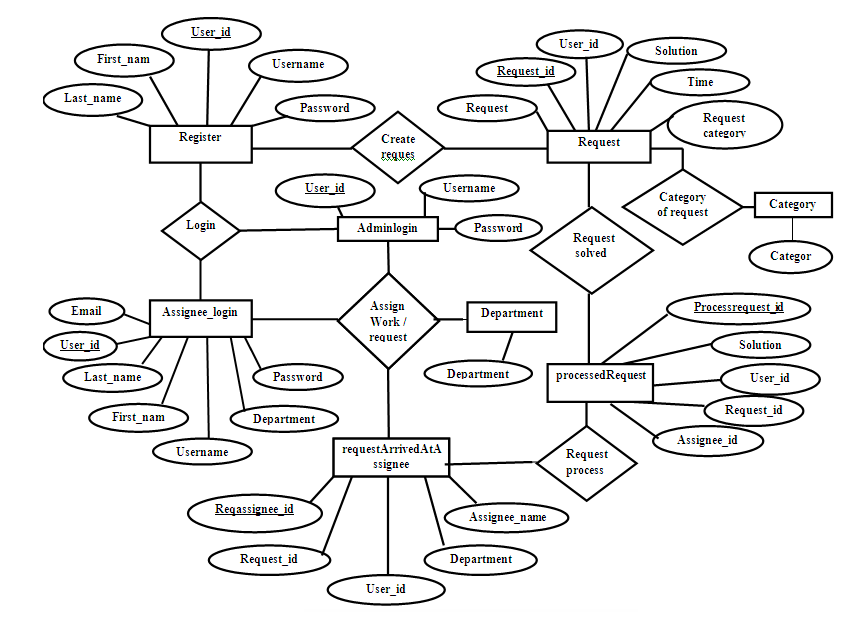
Administrator Use Case

Admin  
(faculty head)

1. **Guest Use Case**

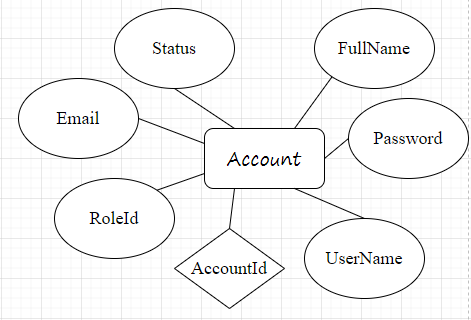
Student  
(user)

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

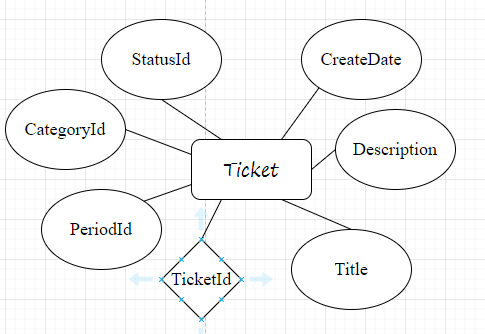


1. **Entities and Properties**

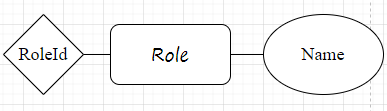
**2.1 Account Entity & Properties**

****

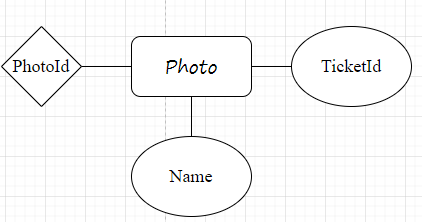
**2.2 Ticket Entity & Properties**



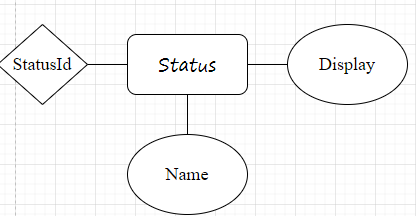
**2.3 Role Entity & Properties**



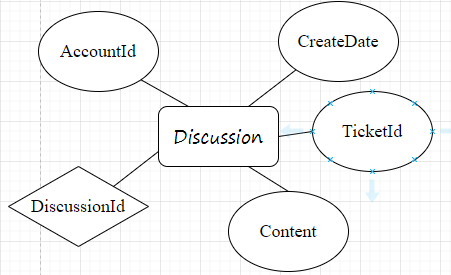
**2.4 Photo Entity & Properties**



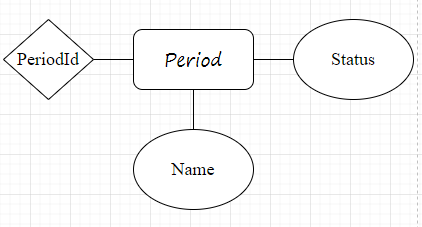
**2.5 Status Entity & Properties**



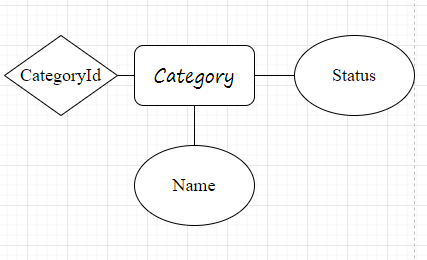
**2.6 Discussion Entity & Properties**



**2.7 Period Entity & Properties**



**2.8 Category Entity & Properties**

****

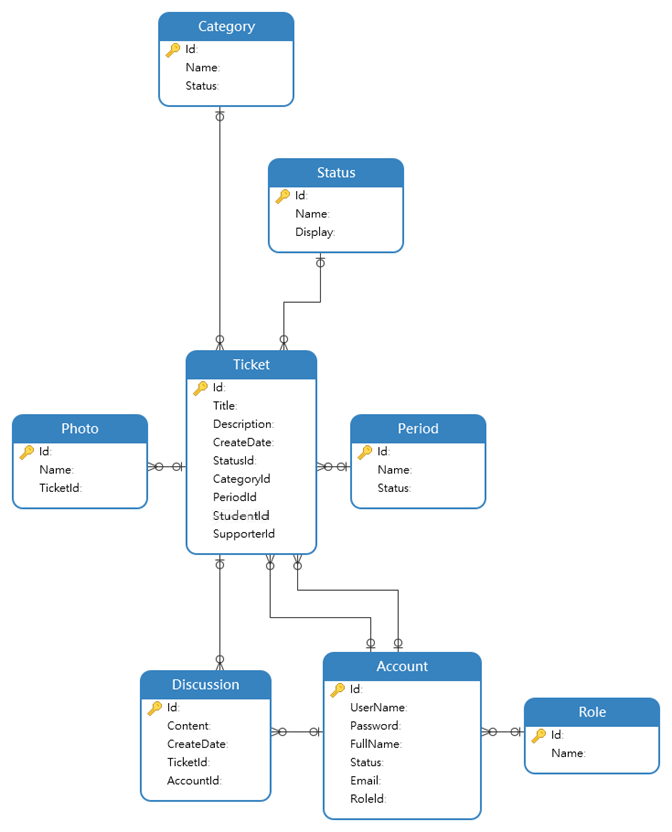
**Task sheet review 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | Jan 18, 2021 | 6 | All Members | Complete |
| 02 | Algorithms - Data Flowchart | Nguyen Minh Thoai | Complete |
| 03 | Data Flow Diagram | Nguyen Minh Thoai | Complete |
| 04 | Use Case Diagram | Hoang Thien An | Complete |
| 05 | ERD | Hoang Thien An | Complete |

|  |  |  |
| --- | --- | --- |
|  | Prepare By: Group 4 |  |
| Date:Jan 18, 2021 | Team Leader  Nguyen Minh Thoai | Mr. Ngo Phuoc Nguyen |

**REVIEW III**

1. Database Design
2. Database Design Diagram



1. Database Structure

2.1 Table Account

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| Id | int | Not Null |  | PK |  | Field ID |
| UserName | varchar | No |  |  |  | Account for Login |
| Password | varchar | No |  |  |  | Password for Login |
| FullName | varchar | No |  |  |  | Name of User |
| Status | bit | No |  |  |  | Status checked |
| Email | varchar | No |  |  |  | Email User |
| Roleld | int | No |  |  |  | Field ID |
| Phone | Int | No |  |  |  | Number Phone |

**2.2 Table Category**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| Id | int | Not Null |  | PK |  | Field ID |
| Name | varchar | No |  |  |  | Name of Category |
| Status | varchar | No |  |  |  | Check Active or Unactive |

**2.3 Table Discusssion**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| Id | int | Not Null |  | PK |  | Field ID |
| Content | text | No |  |  |  | Description |
| CreateDate | date | No |  | PK |  | Date Created |
| TicketId | int | No |  | FK |  | Field ID |
| AccountId | int | No |  | FK |  | Field ID |

**2.4 Table Period**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| Id | int | Not Null |  | PK |  | Field ID |
| Name | varchar | No |  |  |  | Student name |
| Status | bit | No |  |  |  | Check Active or Unactive |
| Color | varchar | No |  |  |  | Color of Period |

**2.5 Table Photo**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| Id | int | Not Null |  | PK |  | Field ID |
| Name | varchar | No |  |  |  | Name of Photo |
| TicketId | int | No |  | FK |  | Field ID |

* 1. Table Role

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| Id | int | Not Null |  | PK |  | Field ID |
| Name | varchar | No |  |  |  | Name of Role |

**2.7 Table Status**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| Id | int | Not null |  | PK |  | Field ID |
| Name | varchar | No |  |  |  | Name of Status |
| Display | bit | No |  |  |  | Display of Status as |
| Color | varchar | No |  |  |  | Color of Status |

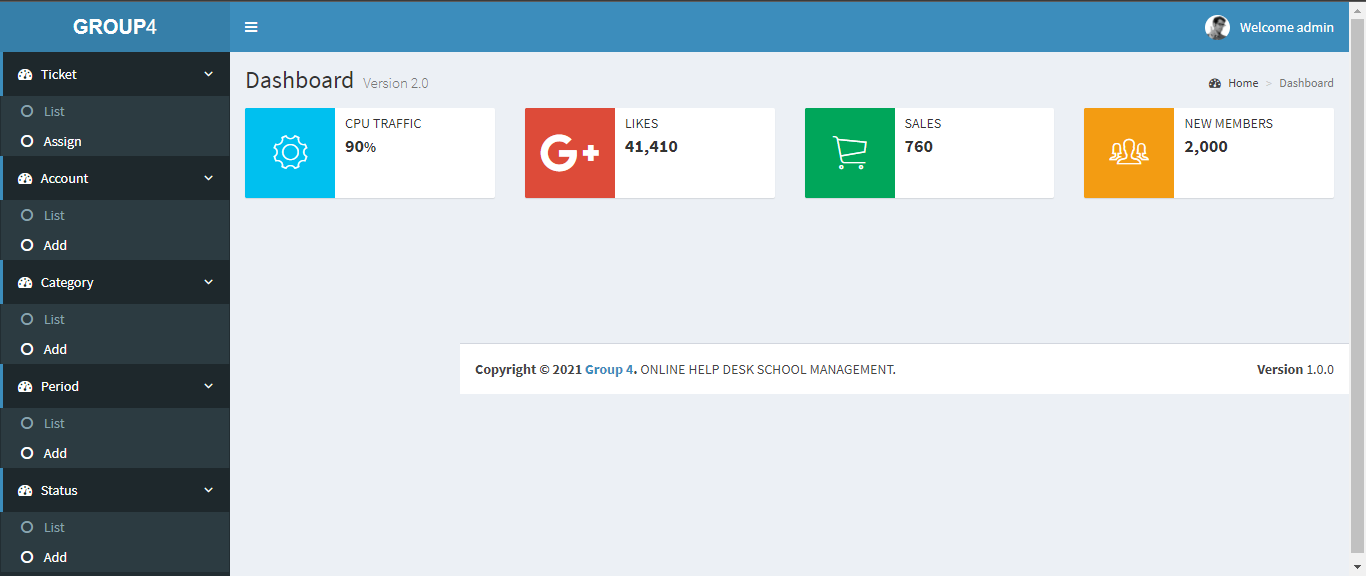
**2.8 Table Ticket**

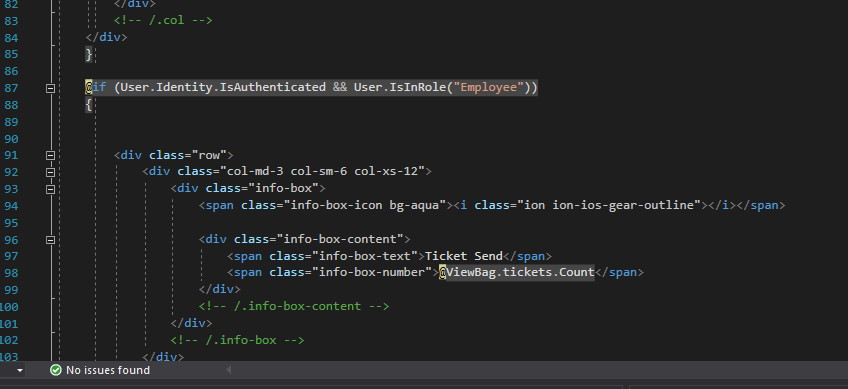
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Null | Default | Key | Field-Ref | Description |
| Id | int | Not null |  | PK |  | Field ID |
| Title | varchar | No |  |  |  | Name of Ticket |
| Description | text | No |  |  |  | About the Ticket |
| CreateDate | date | No |  |  |  | Time Ticket Created |
| StatusId | int | No |  | **FK** |  | Field ID |
| CategoryId | int | No |  | **FK** |  | Field ID |
| PeriodId | int | No |  | **FK** |  | Field ID |
| StudentId | int | No |  | **FK** |  | Field ID |
| SupporterId | int | No |  | **FK** |  | Field ID |

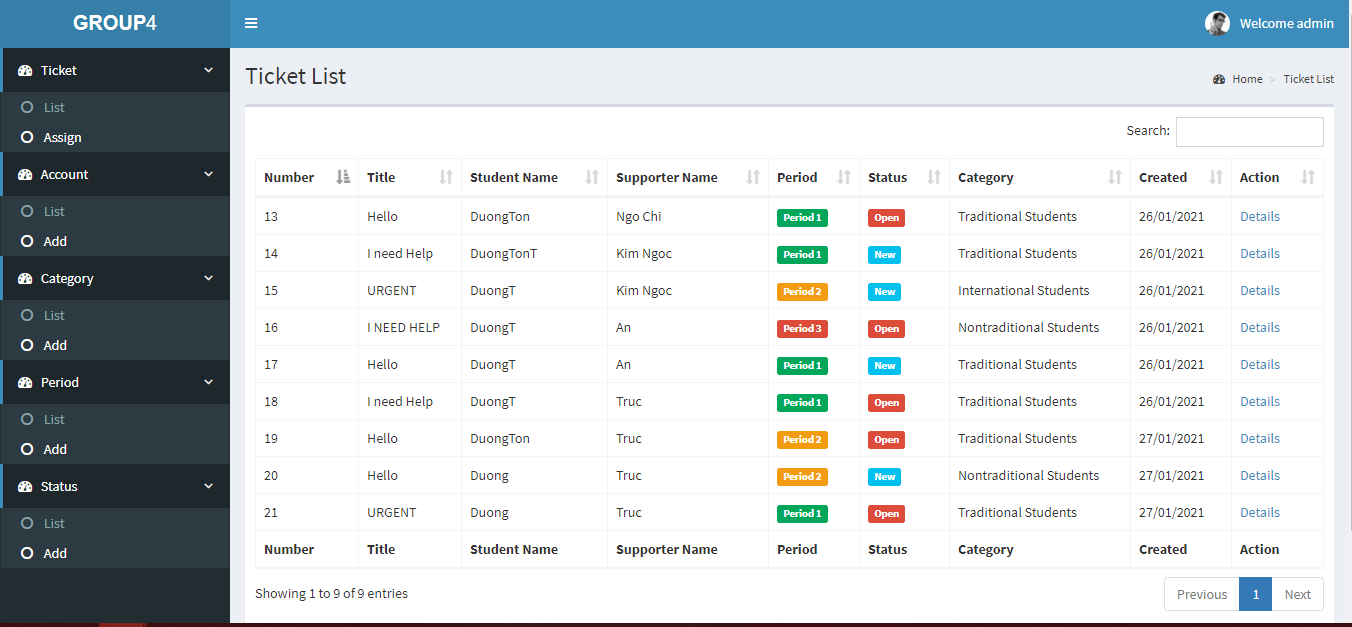
GUI Design – Front-end &

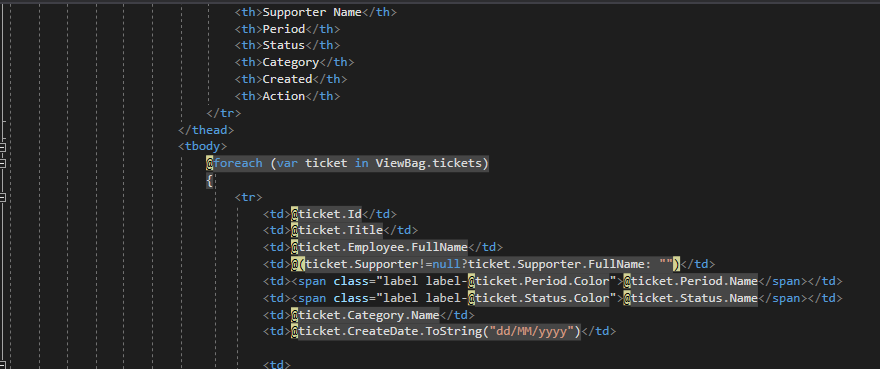
GUI Design – Back-end:

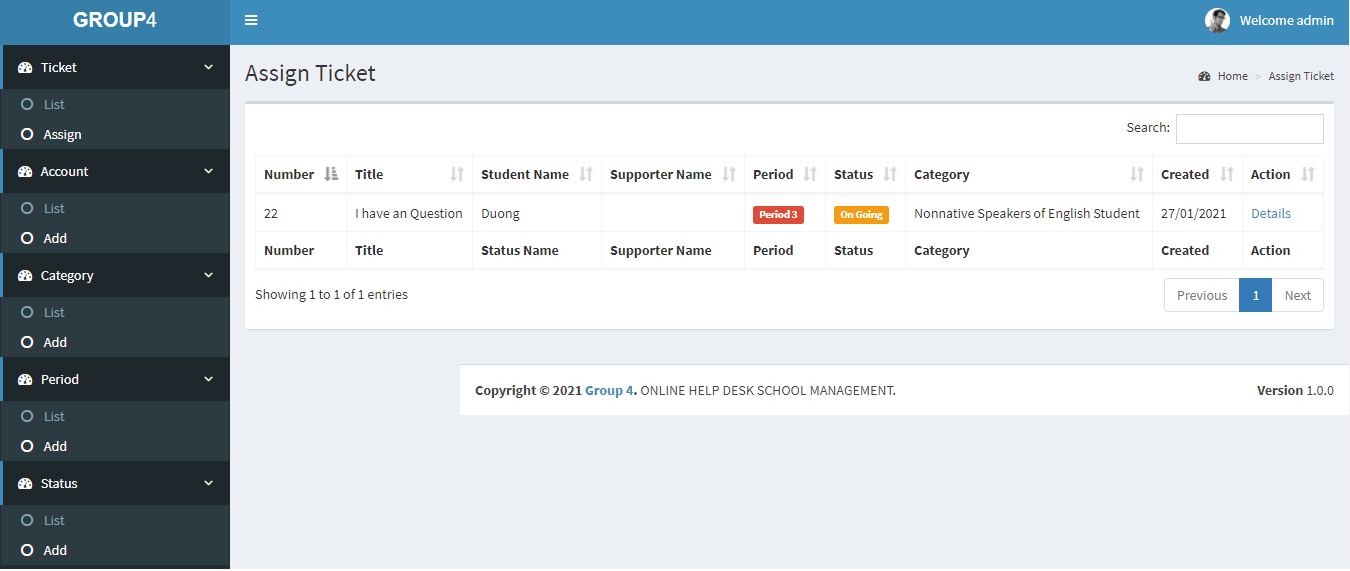
1. Admin Home page:

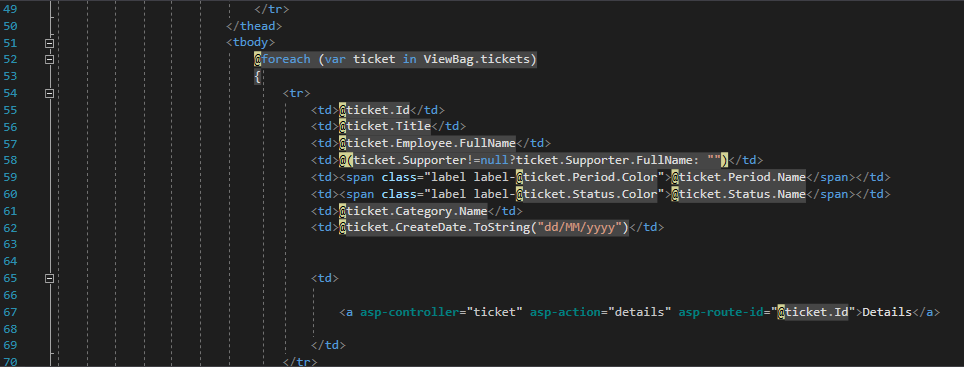


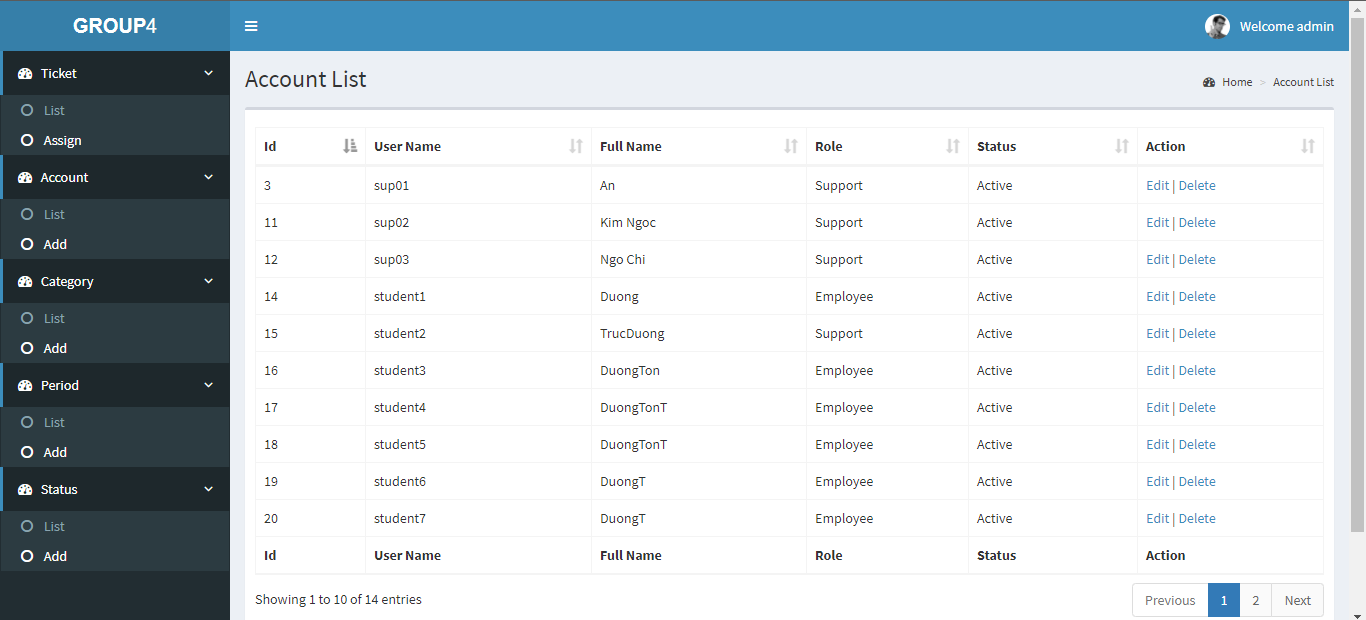


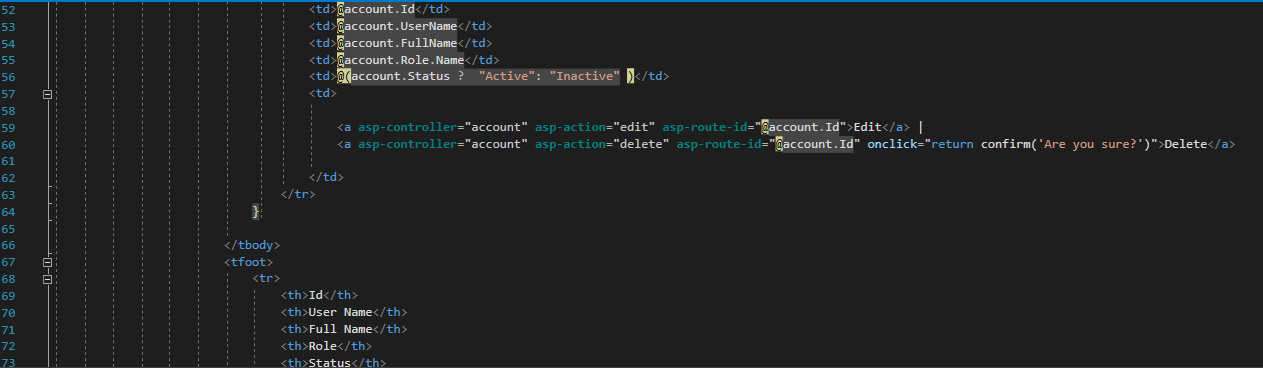


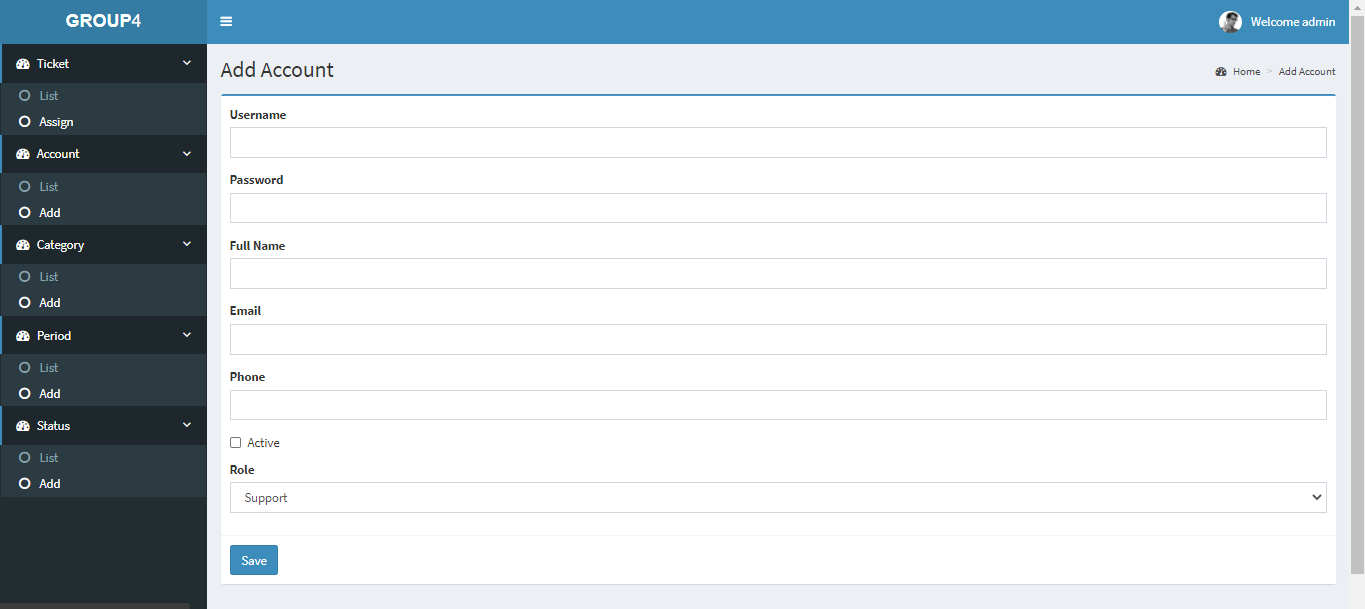


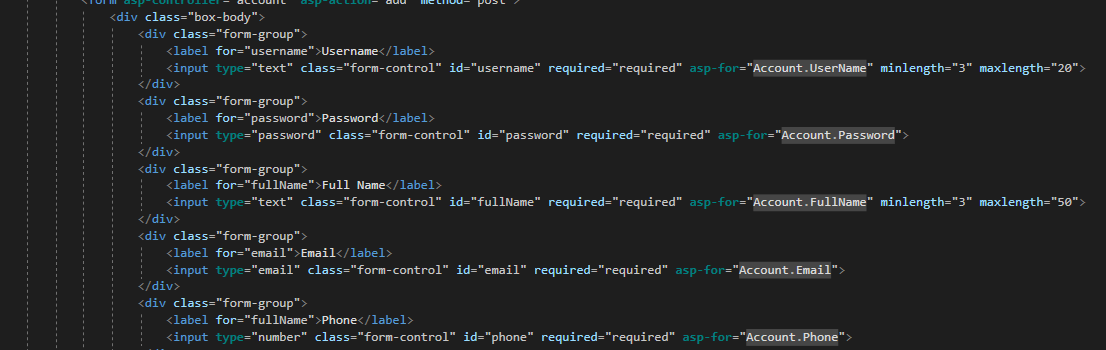


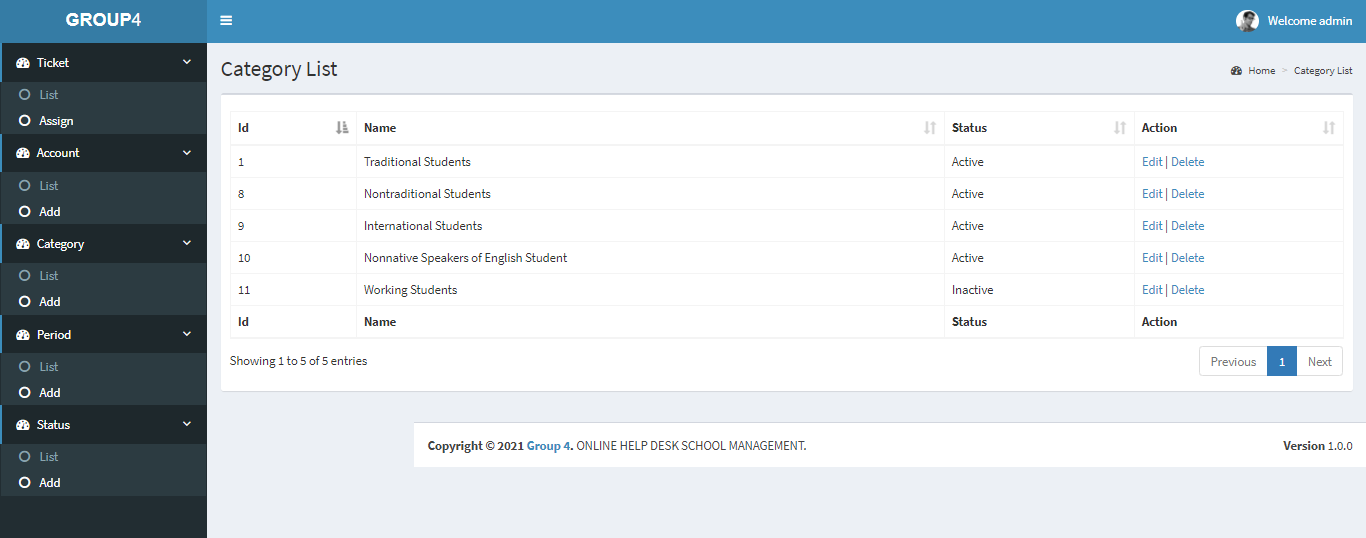


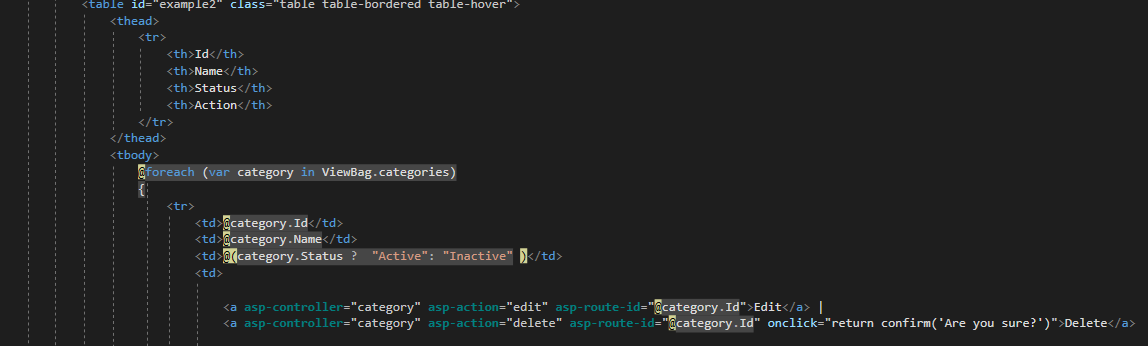


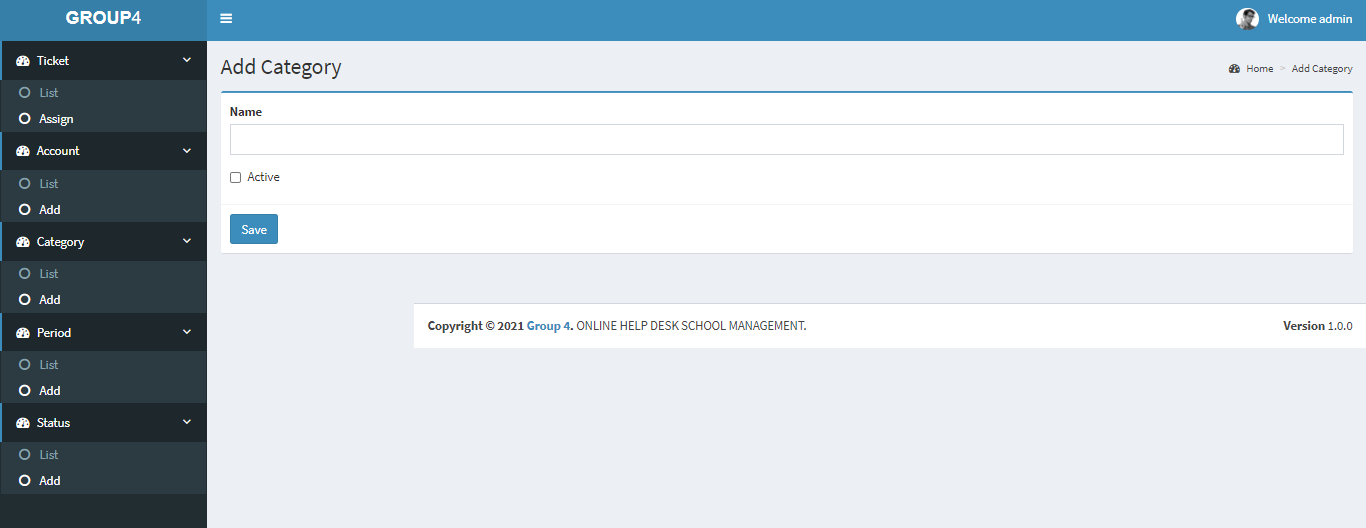




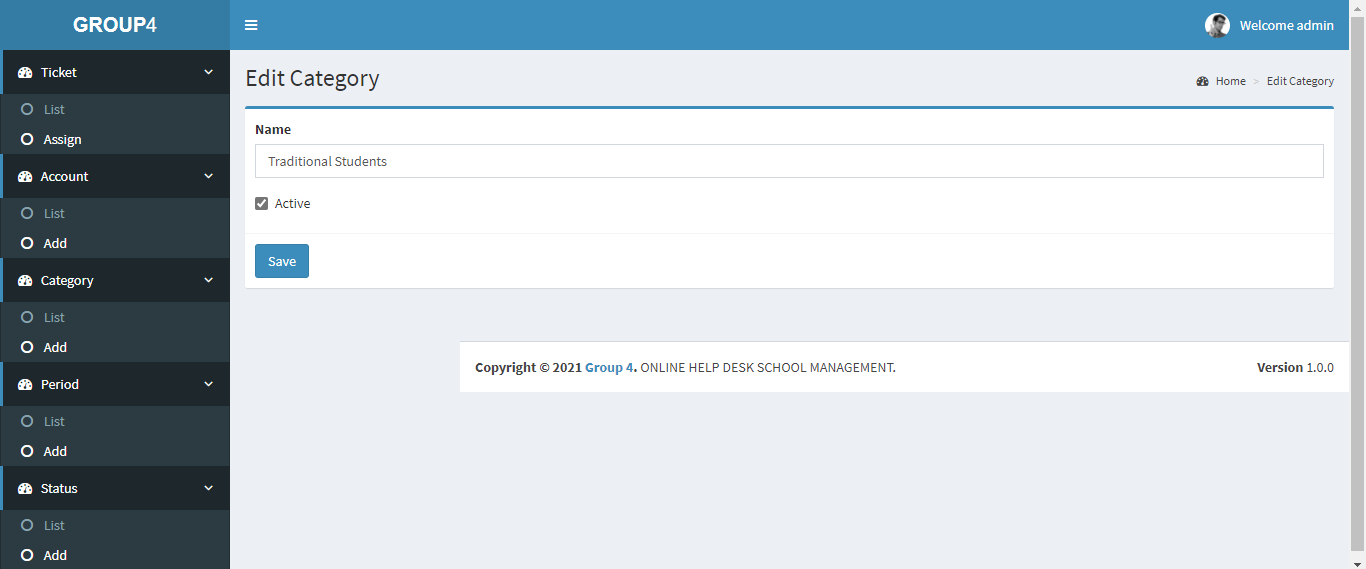




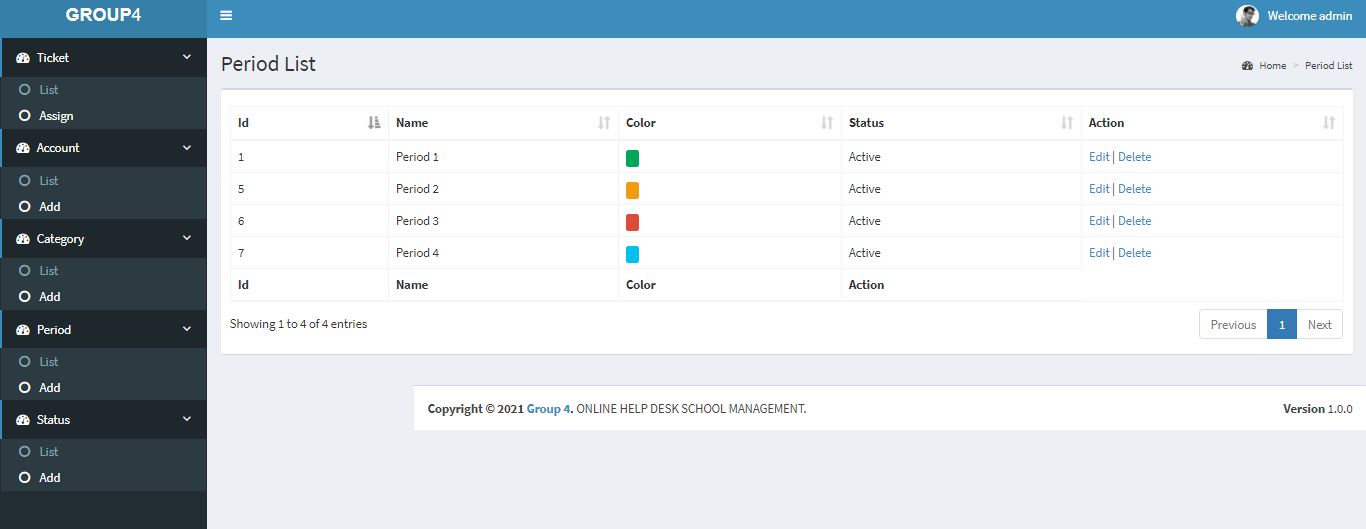


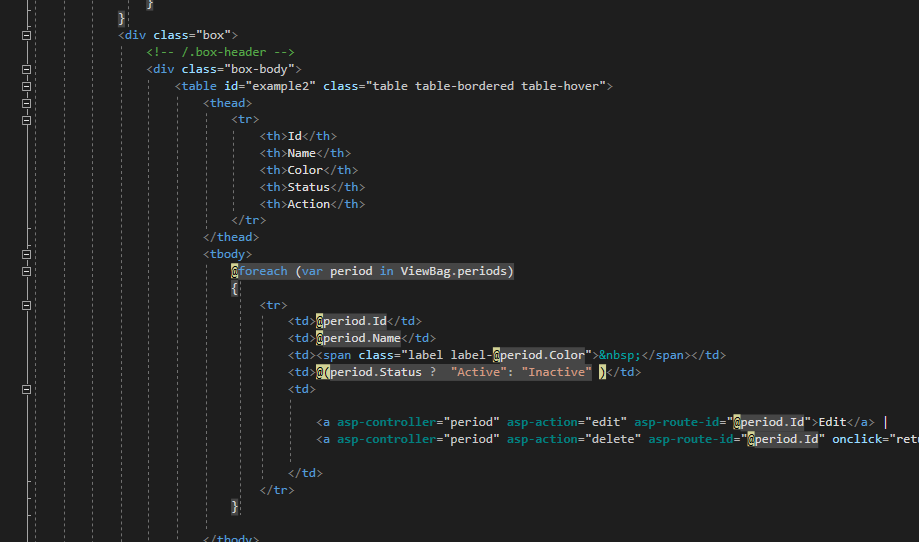


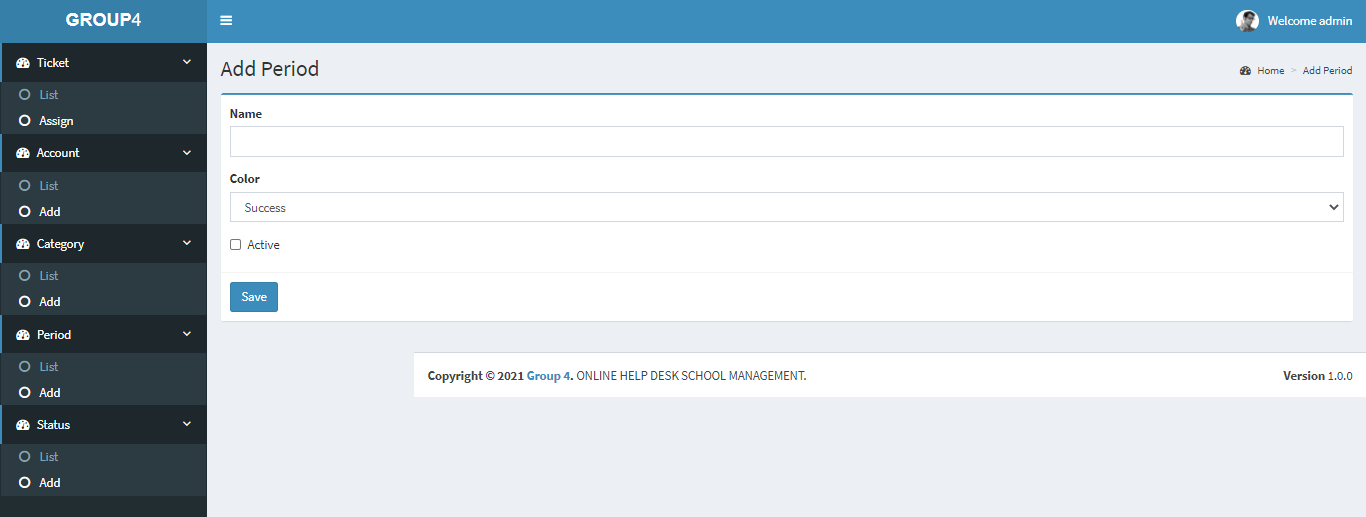


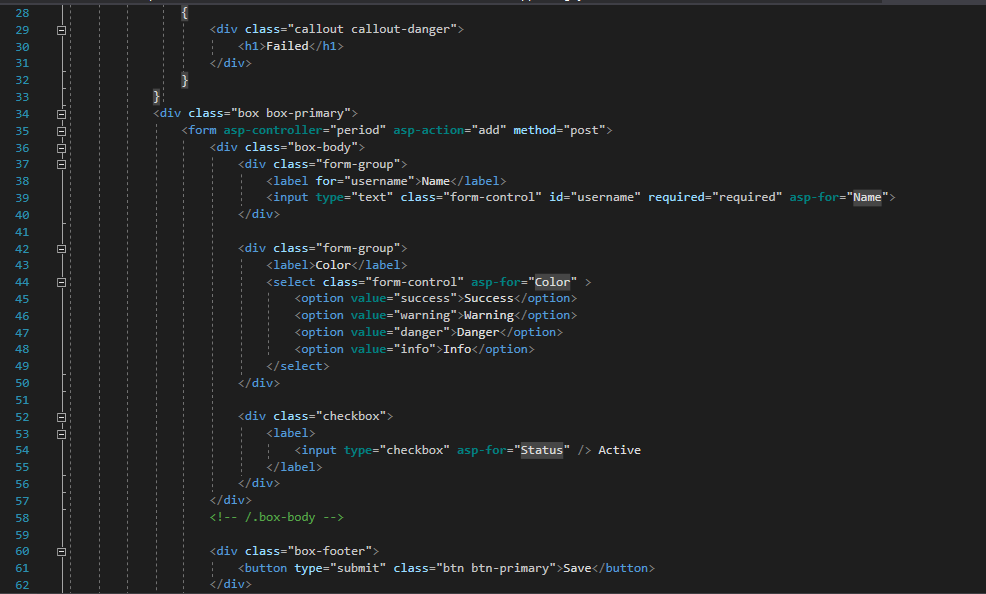


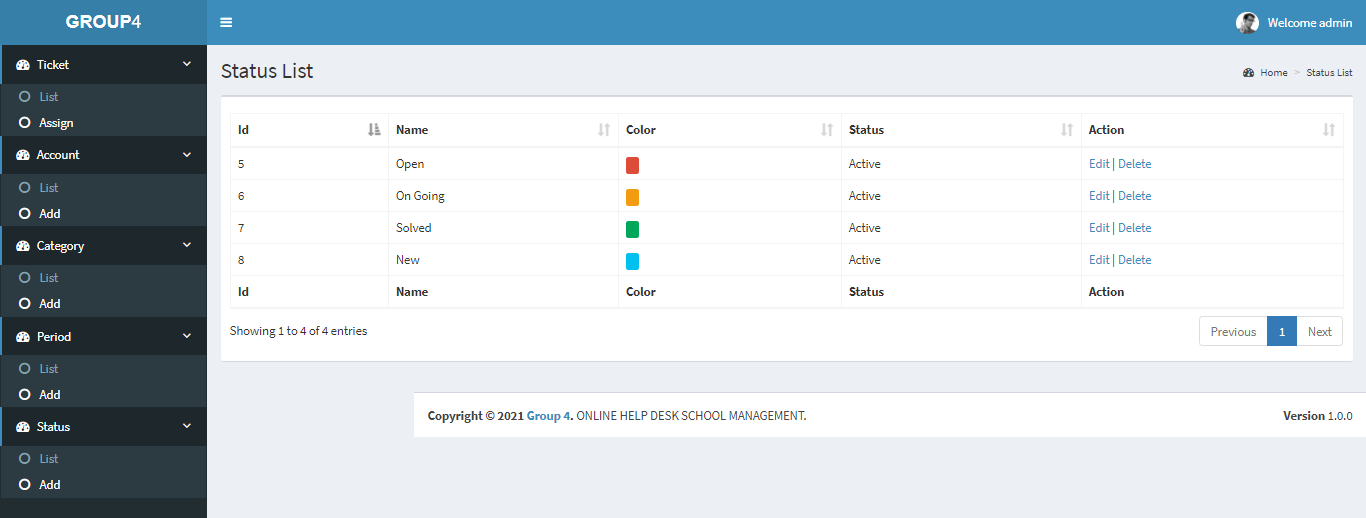


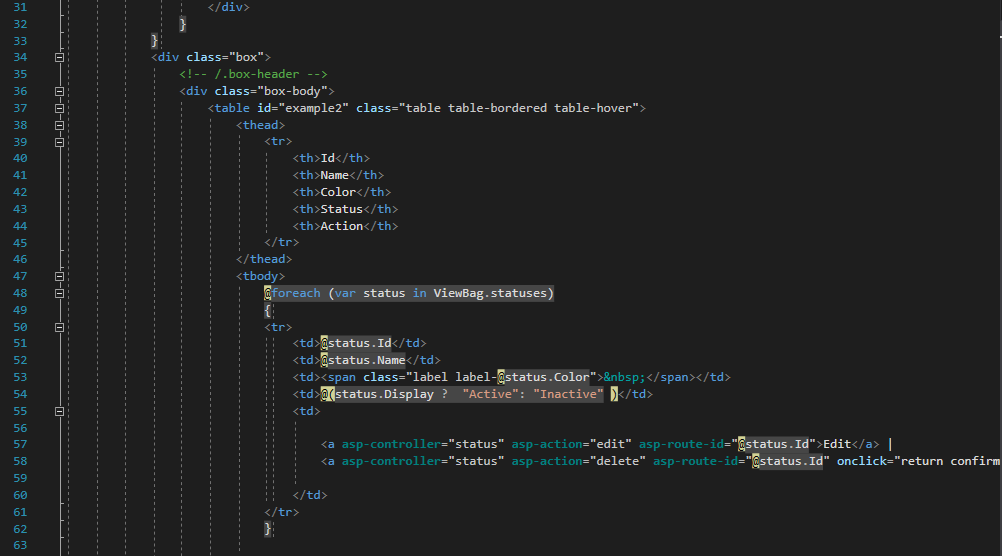


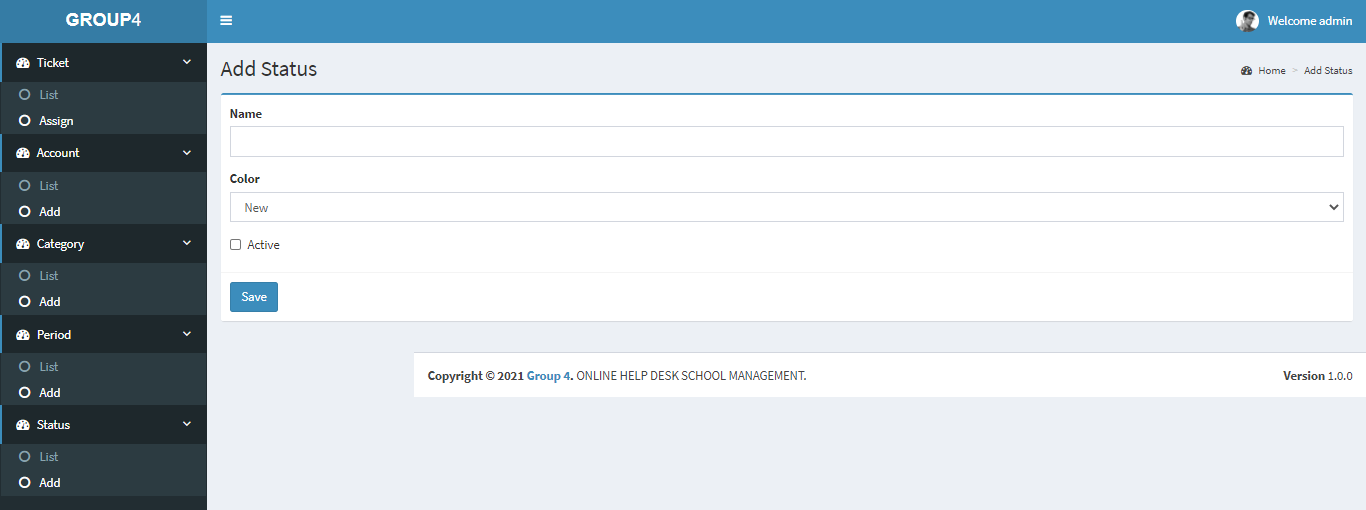


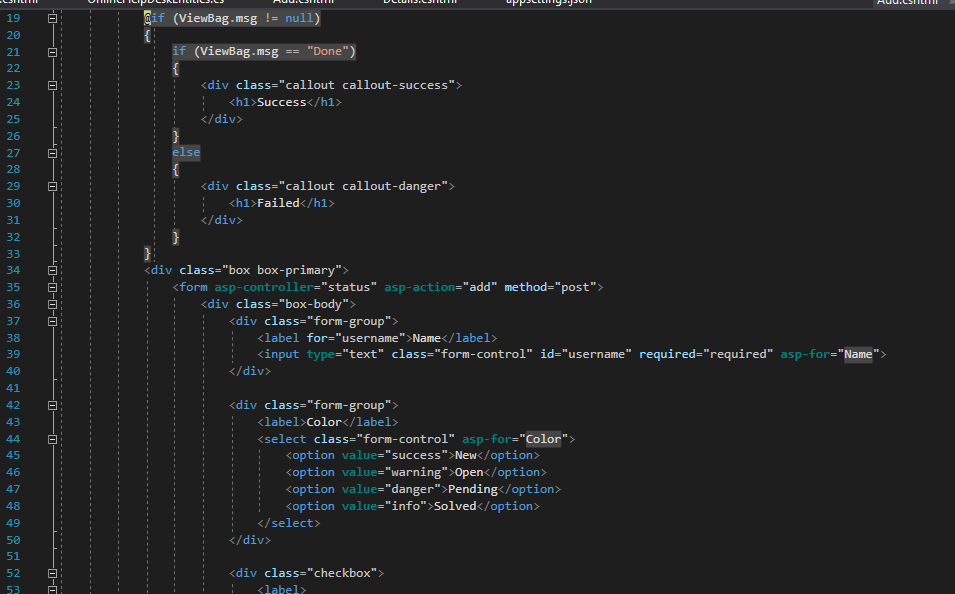






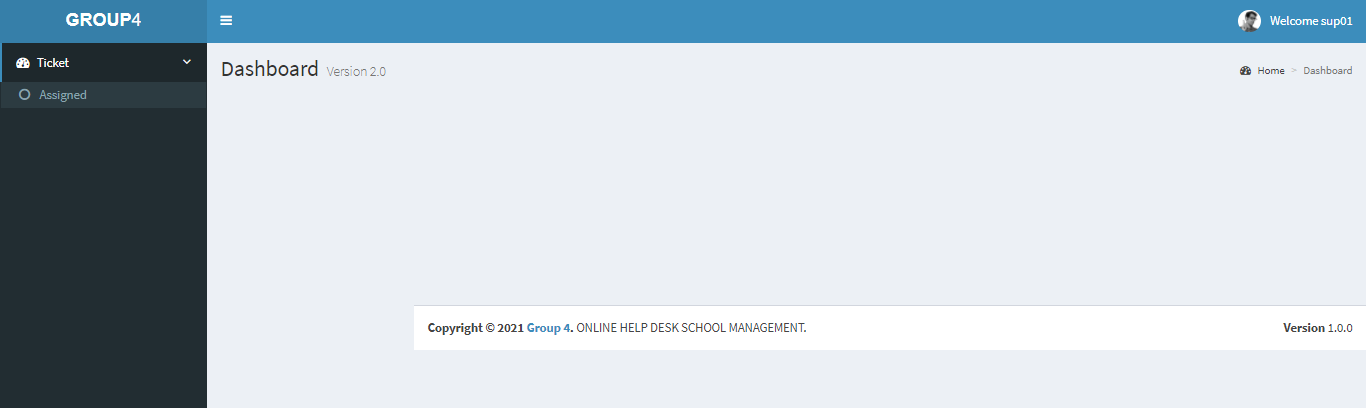


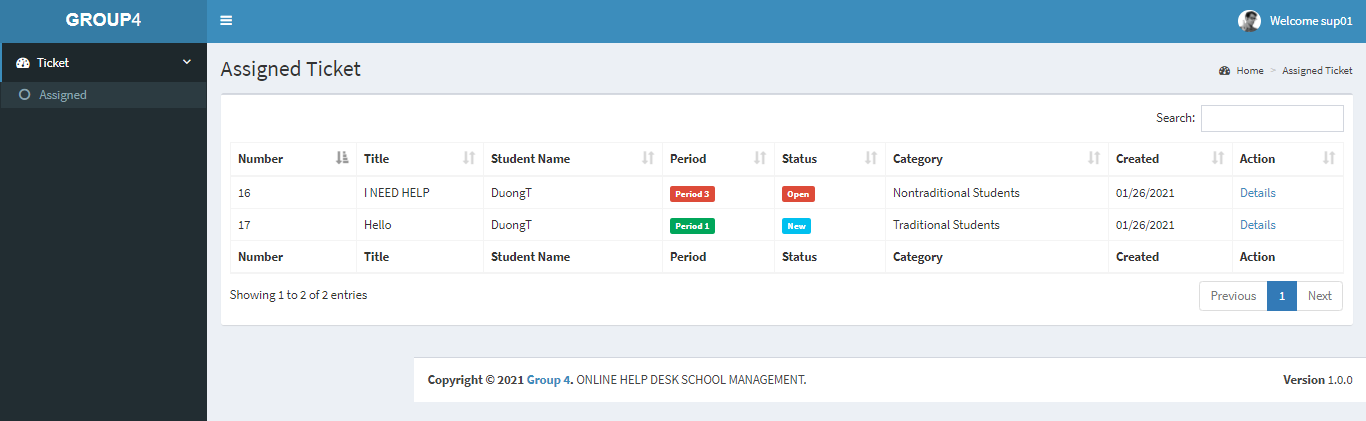


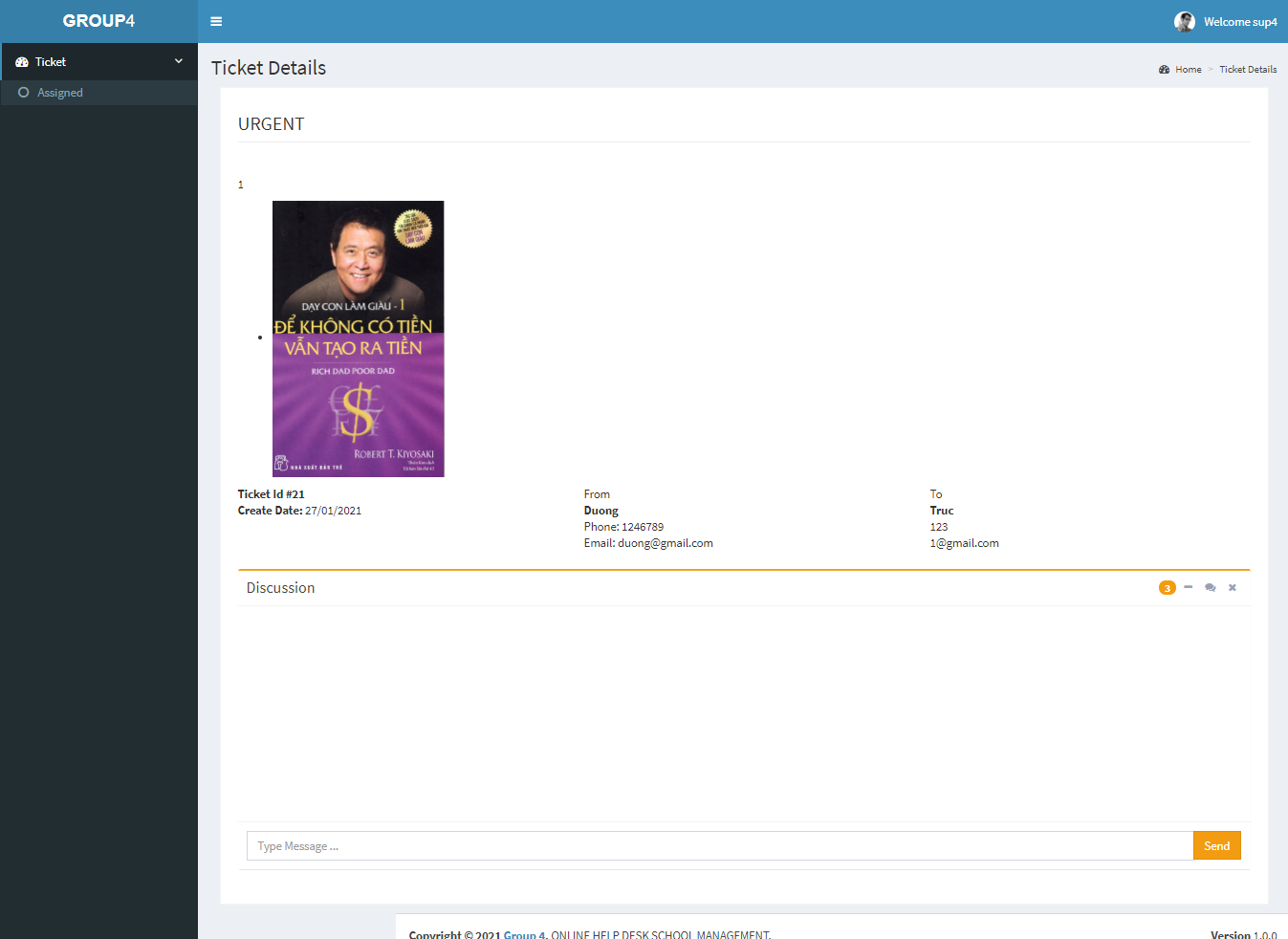


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Admin Page | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | Home | Hyberlink |  | Click | URL to Home | Enable |
| 2 | List | Hyberlink |  | Click | URL to List | Enable |
| 3 | Assign | Hyberlink |  | Click | URL to Assign | Enable |
| 4 | Details | Hyberlink |  | Click | URL to Details | Enable |
| 5 | Assign | Hyberlink |  | Click | URL to Assign | Enable |
| 6 | Details | Hyberlink |  | Click | URL to Details | Enable |
| 7 | Home | Hyberlink |  | Click | URL to Home | Enable |
| 8 | Assign | Hyberlink |  | Click | URL to Assign | Enable |
| 9 | Details | Hyberlink |  | Click | URL to Details | Enable |
| 10 | Discussion | Hyberlink |  | Click | Chat RealTime with Student | Enable |

**2. Suport Home.**

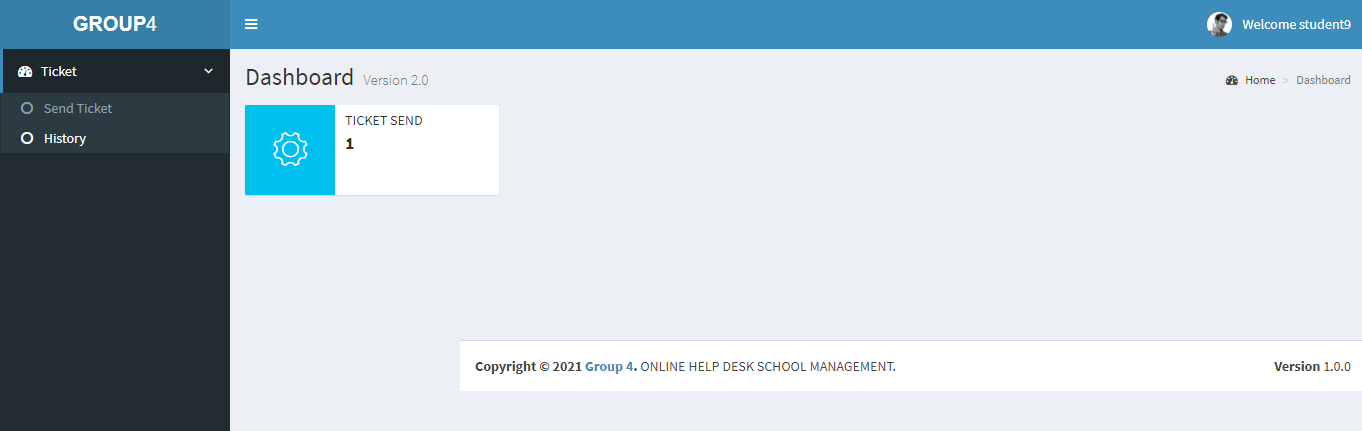
****

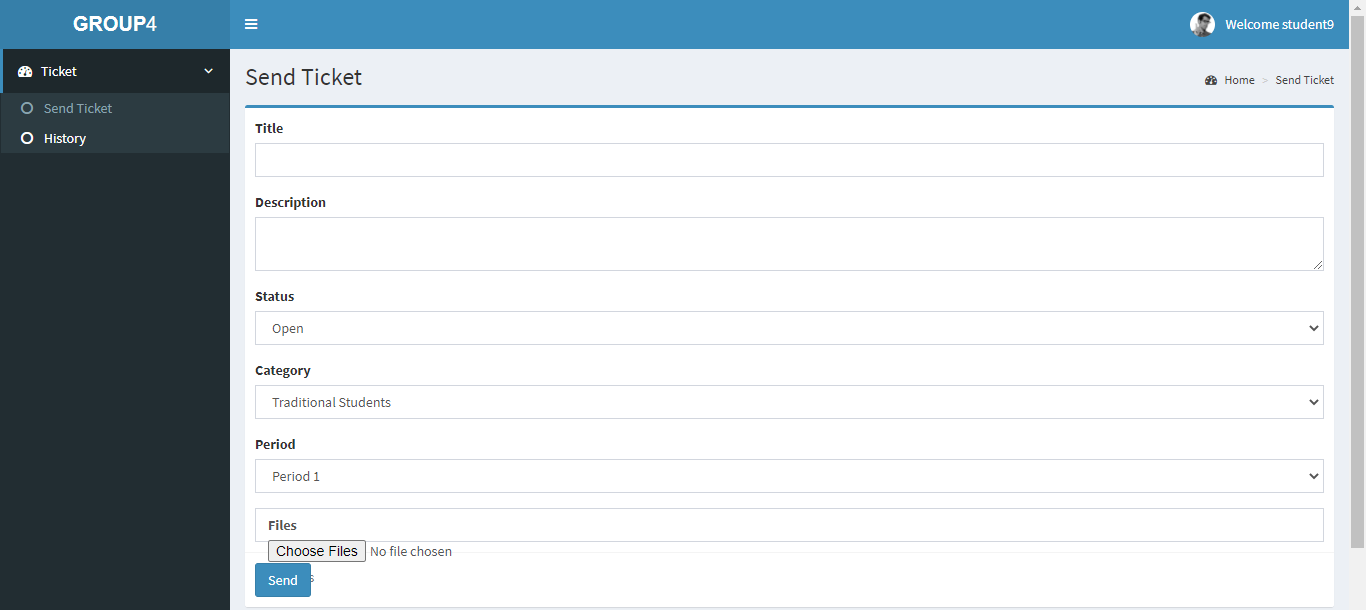
****

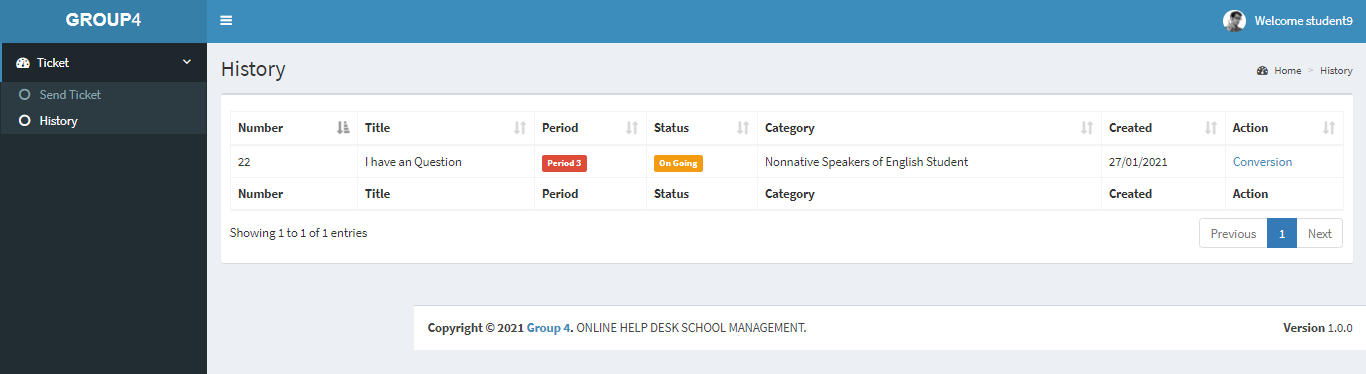


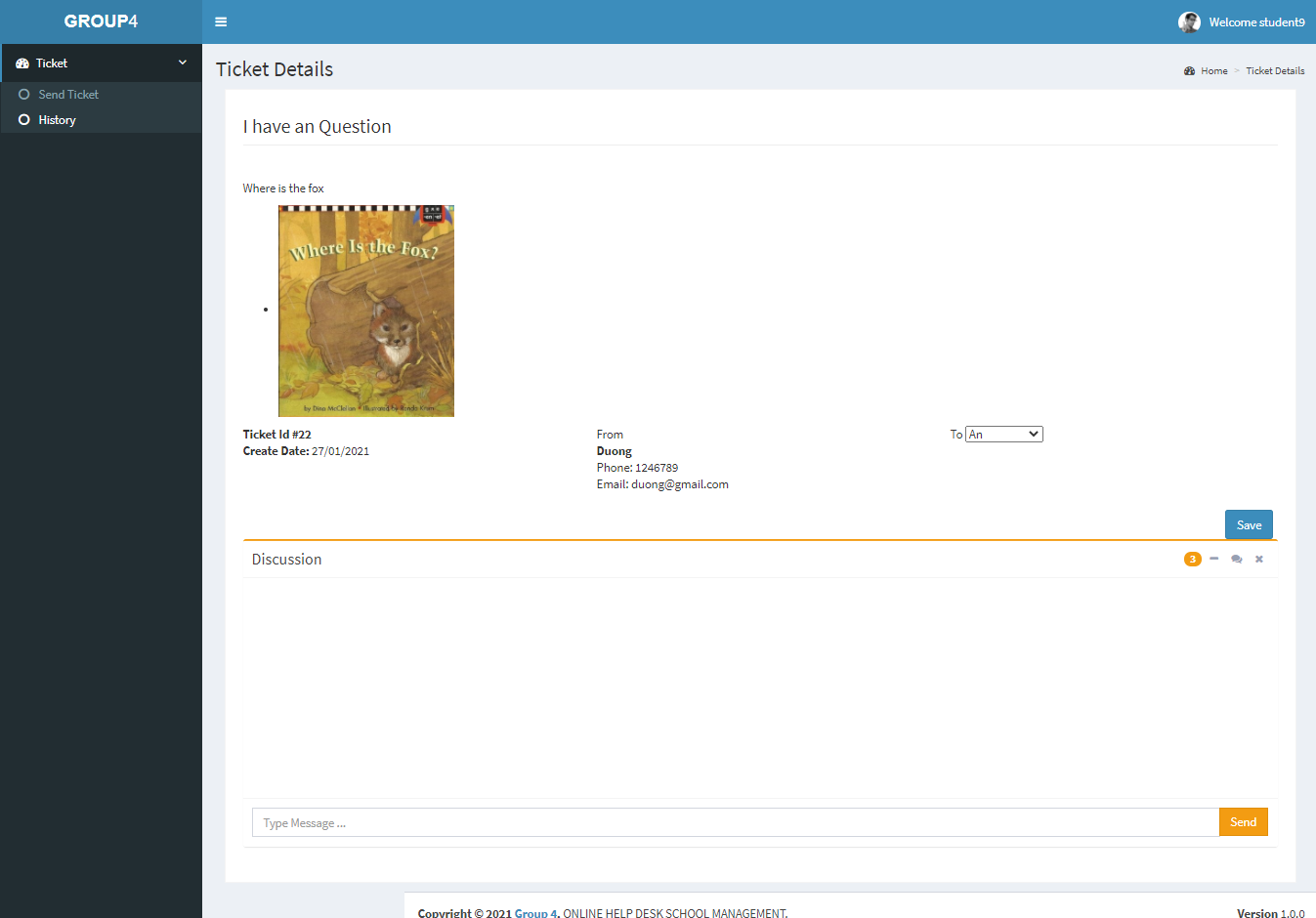
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Support Page | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | Home | Hyberlink |  | Click | URL to Home | Enable |
| 2 | Assign | Hyberlink |  | Click | URL to Assign | Enable |
| 3 | Details | Hyberlink |  | Click | URL to Details | Enable |
| 4 | Discussion | Hyberlink |  | Click | Chat RealTime with Student | Enable |

**3. Student Home Page**

****

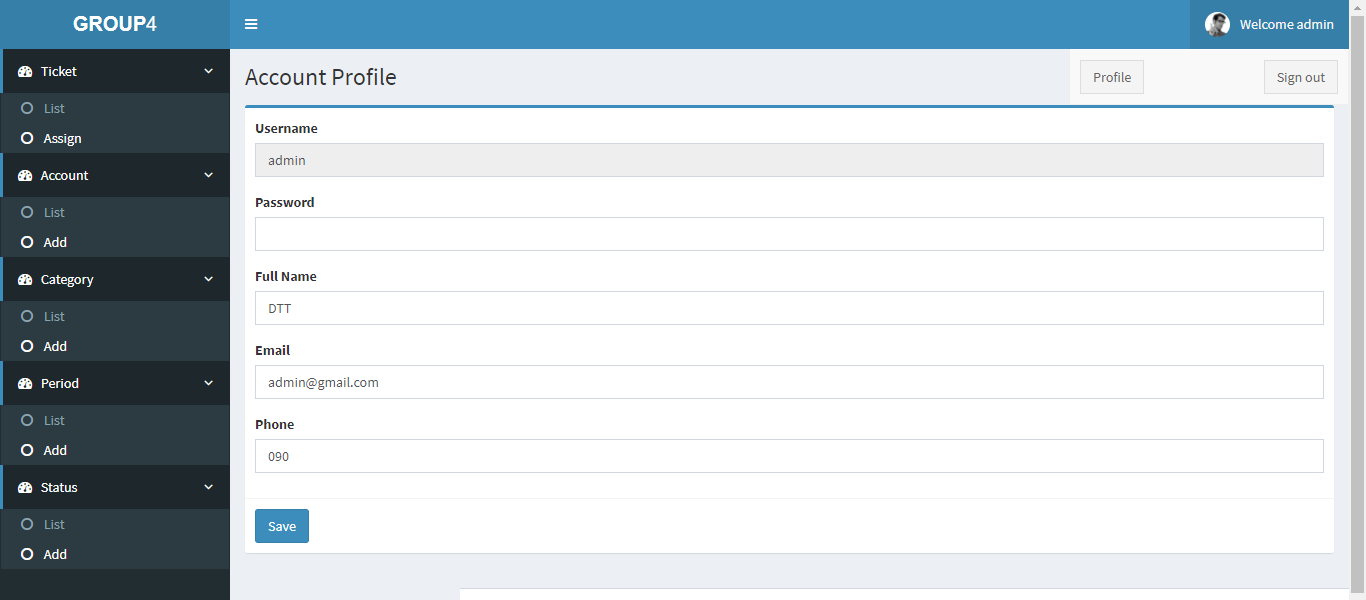
****

****



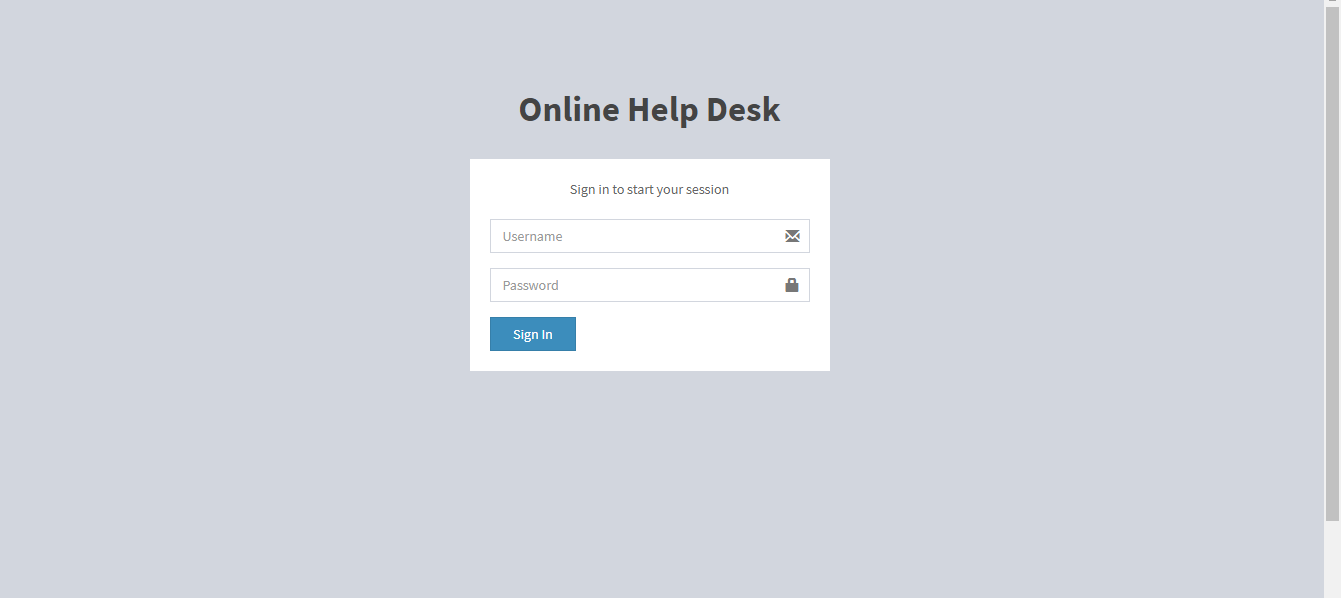
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Student Page | | | | | | |
| No. | Name | Type | Validation | Event | Description | Status |
| 1 | Home | Hyberlink |  | Click | URL to Home | Enable |
| 2 | SendTicket | Hyberlink |  | Click | URL to Send Ticket | Enable |
| 3 | Details | Hyberlink |  | Click | URL to Details | Enable |
| 4 | Discussion | Hyberlink |  | Click | Chat RealTime with Support | Enable |

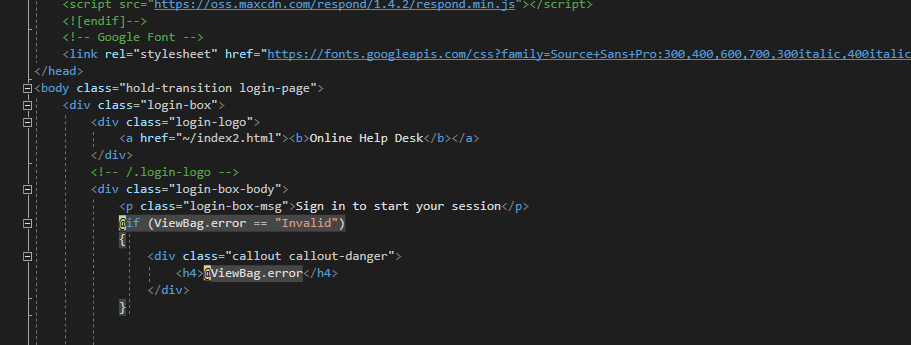
**4. Profile Page**

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Profile Form** | | | | | | |
| No | Name | Type | Validation | Event | Description | Status |
| 1 | UserName | Text box |  | Input | Input student name | Enable |
| 2 | Password | Text box |  | Input | Input password | Enable |
| 3 | Full Name | Text box |  | Input | Input name | Enable |
| 4 | Email | Text box |  | Input | Input student email | Enable |
| 5 | Phone | Text box |  | Input | Input phone number | Enable |

**5. Login Form**

****

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Login form** | | | | | | |
| No | Name | Type | Validation | Event | Description | Status |
| 1 | UserName | Text box |  | Input | Input admin id | Enable |
| 2 | Password | Text box |  | Input | Input password | Enable |
| 3 | btnSign In | Button |  | Click | Submit for login | 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 | Jan 18,2021 |  | Duong Ton Truc | Complete |
| 02 | Database Structure | Duong Ton Truc | Complete |
| 03 | GUI Design Front-End | Duong Ton Truc | Complete |
| 04 | GUI Design Back-End | Nguyen Minh Thoai | Complete |

|  |  |  |
| --- | --- | --- |
|  | Prepare By: Group 4 |  |
| Date: Jan 18,2021 | Team Leader  Nguyen Minh Thoai | Teacher  Mr. Ngo Phuoc Nguyen |