A

PROJECT REPORT ON

**Leave Management System**

**By**

**ANKIT BHARANI (CE-010) (19CEUBD004)**

**BRIJESH CHUDASAMA (CE-026) (18CEUBD009)**

**YASH GARALA (CE-043) (18CEUOS078)**

**B.Tech CE Semester-VI**

**Subject: System Design Practice**

**Guided by:**

Prof.Pandav K. Patel

Assistant Professor

Dept. of Comp. Engg.



**Faculty of Technology**

**Department of Computer Engineering**

**Dharmsinh Desai University**

****

**Faculty of Technology**

**Department of Computer Engineering**

**Dharmsinh Desai University**

**CERTIFICATE**

This is to certify that the practical / term work carried out in the subject of **System Design Practice** and recorded in this journal is the

bonafide work of

**ANKIT BHARANI (CE-010) (19CEUBD004)**

**BRIJESH CHUDASAMA (CE-026) (18CEUBD009)**

**YASH GARALA (CE-043) (18CEUOS078)**

of B.Tech semester **VI** in the branch of **Computer Engineering** during the academic year **2020-2021**.

Prof. Pandav K. Patel Dr. C. K. Bhensdadia,

Assistant Professor, Head,

Dept. of Computer Engg., Dept. of Computer Engg.,

Faculty of Technology Faculty of Technology

Dharmsinh Desai University, Nadiad Dharmsinh Desai University, Nadiad

**TABLE OF CONTENTS**

* **Abstract………………………………………………………..ii**
* **Chapter 1 – Introduction…………………………………… .1**

1.1 Purpose……………………………………… ...1

1.2 Technology used…………………………... ….1

* **Chapter 2 - Software Requirement Specifications…………2**

2.1 Types of user…………………………………..2

2.2 System functional requirements……………….2

2.3 Other non-functional requirements……… ……9

* **Chapter 3 – Design……………………………………… … 10**

3.1 Use Case Diagram……………………… ……10

3.2 Class Diagram…………………………… …..11

3.3 Sequence Diagram……………………… …...12

3.4 State Diagram………………………… ……..14

3.5 Activity Diagram………………………… ….15

3.6 E-R Diagram………………………………. ..17

3.6 Data Dictionary…………………………… . .18

* **Chapter 4 - Implementation Detail……………………. .…20**

4.1 Brief description of modules…………………20

* **Chapter 5 - Test Case Design………………………………21**
* **Chapter 6 – Screenshots…………………………………….22**
* **Chapter 7 - Conclusion and Future Extension……………26**

7.1 Conclusion……………………………………26

7.2 Future Extension……………………………...26

* **Chapter 9 – Bibliography…………………………………. .27**

**Abstract**

Online leave management system is an application that is of utmost importance for an organization or college. The leave management application will allow the users to apply for the leaves through the online mode by specifying a valid reason and valid type for the leave. This can also be implemented at the organization level and the college levels where the employees can apply for the leave. The user interface must be simple and easy to understand. The leave applications and the approvals are done manually. This work can be automated through the use of this application. This application can also allow the users to check the pending leaves that they have.

This application can help in avoiding the leave application that needs to be given to the higher authority with great ease. This application can also help in sending the email notifications to the users regarding the leaves that they have availed with great ease.

**Chapter 1 – Introduction**

1. **Purpose**

Online leave management system is a very handy tool to overcome many manual tasks related to the leave management into organization like university or company.

Authorities just have to create any leave type and it will allocated to all the employees are working inside. Any user can request for leave on the web platform without making any manual task such as going into authorities and fill the forms and etc tasks will be removed.

So, it’s a bridge made of technology that manage Leave management task of users.

1. **Technology Used**

For the implementation, we have used asp.net MVC EF Core framework and C# language as a backend.

Development Tools used during complete implementation of the project is Visual Studio and Umlet.

**Chapter 2 - Software Requirements Specification**

**2.1 Types of User:**

1. Admin
2. Registrar
3. Dean
4. HOD
5. Faculty
6. Pending

**2.2 System Functional Requirements**

**R.1 Admin**

**R.1.1 Login in System**

**Description:** Admin can login to the website to perform leave related operations.

**Input:** Admin click on submit button.

**Output:** Success message.

**R.1.2 Create Leave type**

**Description:** This function allows admin to create Leave type and Number of the days of the leave.

**Input:** Provide proper name of the leave.

**Output:** Success message.

**R.1.3 List out Leave type**

**Description:** This will allow admin to List all the existing Leave type in the system and

**Input:** Provide proper name of the leave.

**Output:** Success message.

**R.1.3.1 Edit Leave type**

**Description:** It will allow to modify leave details according to the requirement.

**Input:** Provide Updated Details and press Update Button

**Output:** You can see updated data on the Leave list page.

**R.1.3.2 Delete Leave type**

**Description:** It will delete particular Leave type.

**Input:** press delete button for respective Leave Type.

**Output:** Leave Deleted Successfully.

**R.1.4 Manage All user’s history.**

**Description:** This will manage all the users accounts exists in the system.

**Input:** press all users history button

**Output:** List of all users.

**R.1.5 Manage All user’s Leave history.**

**Description:** This will manage all the users leave request details exists in the system.

**Input:** press Leave history button

**Output:** List of all users Leave history.

**R.1.6 Respond to the registrar Leave request**

**Description:** This will manage all the leave request of the registrar and generate accept/reject response accordingly.

**Input:** Press accept/reject button

**Output:** List of all users Leave history.

**R.1.7 Leave Operation history**

**Description:** This will list out details regarding accept/reject responses.

**Input:** PressMyLeave Operation button

**Output:** List of all leave operations made in past.

**R.1.8 Role allocation**

**Description:** This will allocate the role to newly created user when it registers and upload its profile details verification manually.

**Input:** Allocate Role and Department

**Output:** Role successfully allocated.

**R.1.9 Update Self Profile Details**

**Description:** This will Allow user to Edit self-profile details.

**Input:** Provide proper information and hit edit button

**Output:** Data updated Successfully.

**R.2 Registrar**

**R.2.1 Login in System**

**Description:** Registrar can login to the website to perform leave related operations.

**Input:** Registrar click on submit button.

**Output:** Success message.

**R.2.2 Create Leave Request**

**Description:** This function allows Registrar to create Leave request and send it to Admin.

**Input:** Provide proper leave information of the leave such as leave type, leave reason, starting date, ending date and any attachment file

**Output:** Leave request created successfully.

**R.2.3 Respond to the Dean’s Leave request**

**Description:** This will manage all the leave request of the Deans and generate accept/reject response accordingly.

**Input:** Press accept/reject button

**Output:** Leave response created successfully.

**R.2.4 Manage All Dean’s remaining Leaves.**

**Description:** This will manage all the deans remain leaves exists in the system.

**Input:** press Users Leaves button.

**Output:** List of all user’s remain leaves.

**R.2.5 Leave Operation history**

**Description:** This will list out details regarding accept/reject responses.

**Input:** PressMyLeave Operation button

**Output:** List of all leave operations made in past.

**R.2.6 Update Self Profile Details**

**Description:** This will Allow user to Edit self profile details.

**Input:** Provide proper information and hit edit button

**Output:** Data updated Successfully.

**R.2.7 Manage Self Leave requests made to registrar.**

**Description:** This will manage all the Leave request made to the Admin.

**Input:** press My Leave button.

**Output:** details regarding My leaves.

**R.2.7.1 Edit Leave request**

**Description:** It will allow to modify leave request and resend with proper attachments.

**Input:** Provide Updated Details and press Update Button

**Output:** You can see updated data on the Leave list page

**R.2.7.2 Cancel Leave request**

**Description:** It will allow to registrar cancel the leave requests.

**Input:** Select leave and hit delete button

**Output:** You can see updated data on the Leave list page

**R.3 Dean**

**R.3.1 Login in System**

**Description:** Dean can login to the website to perform leave related operations.

**Input:** Dean click on submit button.

**Output:** Success message.

**R.3.2 Create Leave Request**

**Description:** This function allows Dean to create Leave request and send it to Registrar.

**Input:** Provide proper leave information of the leave such as leave type, leave reason, starting date, ending date and any attachment file

**Output:** Leave request created successfully.

**R.3.3 Respond to the HOD’s Leave request**

**Description:** This will manage all the leave request of the HODs and generate accept/reject response accordingly.

**Input:** Press accept/reject button

**Output:** Leave response created successfully.

**R.3.4 Manage All HOD’s remaining Leaves.**

**Description:** This will manage all the HOD’s remain leaves exists in the system.

**Input:** press Users Leaves button.

**Output:** List of all user’s remain leaves.

**R.3.5 Leave Operation history**

**Description:** This will list out details regarding accept/reject responses.

**Input:** PressMyLeave Operation button

**Output:** List of all leave operations made in past.

**R.3.6 Update Self Profile Details**

**Description:** This will Allow user to Edit self-profile details.

**Input:** Provide proper information and hit edit button.

**Output:** Data updated Successfully.

**R.3.7 Manage Self Leave requests made to registrar.**

**Description:** This will manage all the Leave request made to the Registrar.

**Input:** press My Leave button.

**Output:** details regarding My leaves.

**R.3.7.1 Edit Leave request**

**Description:** It will allow to modify leave request and resend with proper attachments.

**Input:** Provide Updated Details and press Update Button

**Output:** You can see updated data on the Leave list page

**R.3.7.2 Cancel Leave request**

**Description:** It will allow to Dean cancel the leave requests.

**Input:** Select leave and hit delete button

**Output:** You can see updated data on the Leave list page

**R.4 HOD**

**R.4.1 Login in System**

**Description:** HOD can login to the website to perform leave related operations.

**Input:** HOD click on submit button.

**Output:** Success message.

**R.4.2 Create Leave Request**

**Description:** This function allows HOD to create Leave request and send it to Dean.

**Input:** Provide proper leave information of the leave such as leave type, leave reason, starting date, ending date and any attachment file

**Output:** Leave request created successfully.

**R.4.3 Respond to the Faculty’s Leave request**

**Description:** This will manage all the leave request of the Faculty’s and generate accept/reject response accordingly.

**Input:** Press accept/reject button

**Output:** Leave response created successfully.

**R.4.4 Manage All Faculty’s remaining Leaves.**

**Description:** This will manage all the Faculty’s remain leaves exists in the system.

**Input:** press Users Leaves button.

**Output:** List of all user’s remain leaves.

**R.4.5 Leave Operation history**

**Description:** This will list out details regarding accept/reject responses.

**Input:** PressMyLeave Operation button

**Output:** List of all leave operations made in past.

**R.4.6 Update Self Profile Details**

**Description:** This will Allow user to Edit self-profile details.

**Input:** Provide proper information and hit edit button.

**Output:** Data updated Successfully.

**R.4.7 Manage Self Leave requests made to Dean.**

**Description:** This will manage all the Leave request made to the Dean.

**Input:** press My Leave button.

**Output:** details regarding My leaves.

**R.4.7.1 Edit Leave request**

**Description:** It will allow to modify leave request and resend with proper attachments.

**Input:** Provide Updated Details and press Update Button

**Output:** You can see updated data on the Leave list page

**R.4.7.2 Cancel Leave request**

**Description:** It will allow to Dean cancel the leave requests.

**Input:** Select leave and hit delete button

**Output:** You can see updated data on the Leave list page

**R.5 Faculty**

**R.5.1 Login in System**

**Description:** Faculty can login to the website to perform leave related operations.

**Input:** Faculty click on submit button.

**Output:** Success message.

**R.5.2 Create Leave Request**

**Description:** This function allows Faculty to create Leave request and send it to HOD.

**Input:** Provide proper leave information of the leave such as leave type, leave reason, starting date, ending date and any attachment file

**Output:** Leave request created successfully.

**R.5.3 List remaining Leaves.**

**Description:** This will List all the Faculty’s remaining leaves types exists in the system.

**Input:** press Users Leaves button.

**Output:** List of all users remain leaves.

**R.5.4 Update Self Profile Details**

**Description:** This will Allow user to Edit self-profile details.

**Input:** Provide proper information and hit edit button.

**Output:** Data updated Successfully.

**R.5.6 Manage Self Leave requests made to HOD.**

**Description:** This will manage all the Leave request made to the HOD.

**Input:** press My Leave button.

**Output:** details regarding My leaves.

**R.5.6.1 Edit Leave request**

**Description:** It will allow to modify leave request and resend with proper attachments.

**Input:** Provide Updated Details and press Update Button

**Output:** You can see updated data on the Leave list page

**R.5.6.2 Cancel Leave request**

**Description:** It will allow to Dean cancel the leave requests.

**Input:** Select leave and hit delete button

**Output:** You can see updated data on the Leave list page

**R.6 Pending**

**R.6.1 Login in System**

**Description:** Pending role user can login to the website to update the profile details for the verification.

**Input:** Faculty click on submit button.

**Output:** Success message.

**R.6.2 Update Self Profile Details**

**Description:** This will Allow user to Edit self-profile details.

**Input:** Provide proper information and hit edit button.

**Output:** Data updated Successfully.

**R.6.3 Register in System**

**Description:** Pending role user can Register to the website.

**Input:** User provide proper information and press register button.

**Output:** Success message.

**2.3 Other non-functional requirements**

1. **Performance**

The system must be interactive and the delays involved must be less. So, in every action- response of the system, there are no immediate delays. In case of opening App components, of popping error messages and saving the settings or sessions there is delay much below 3 seconds.

1. **Safety**

User details should be securely stored to the server. The main security concern is for user account hence proper login mechanism should be used to avoid hacking.

1. **Reliability**

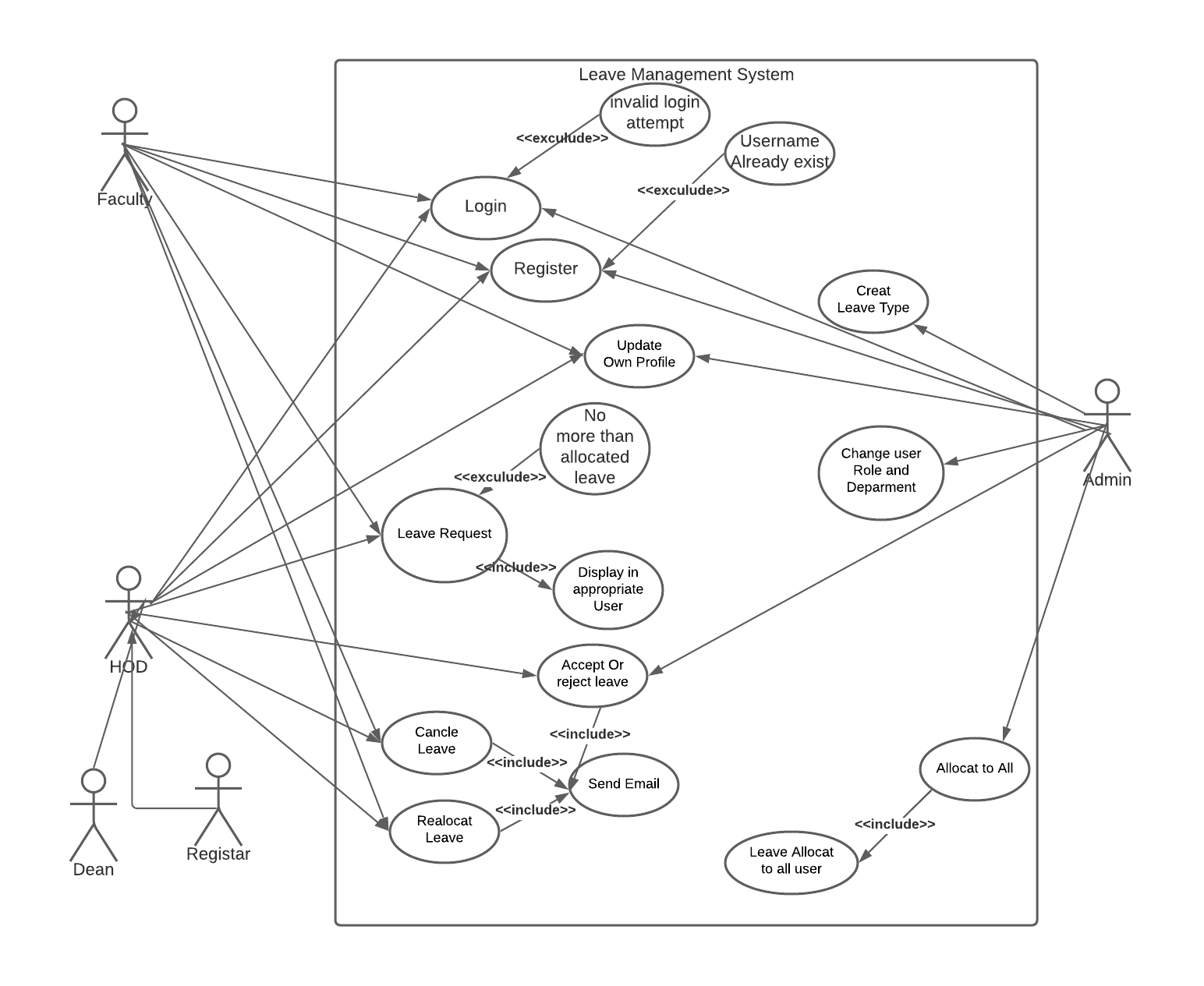
As the system provides the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.

1. **Database**

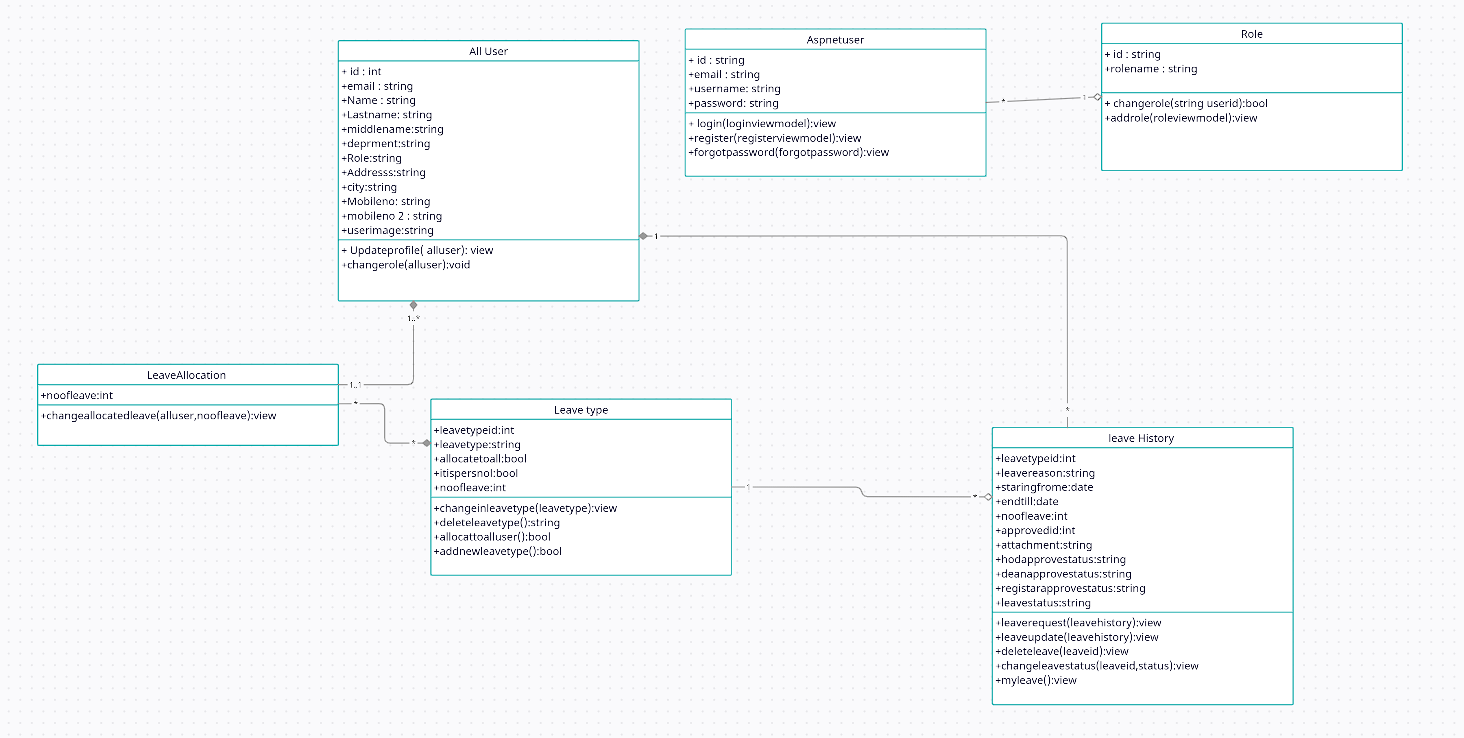
System requires to access user’s data fast to maintain the performance.

**Chapter 3 – Design**

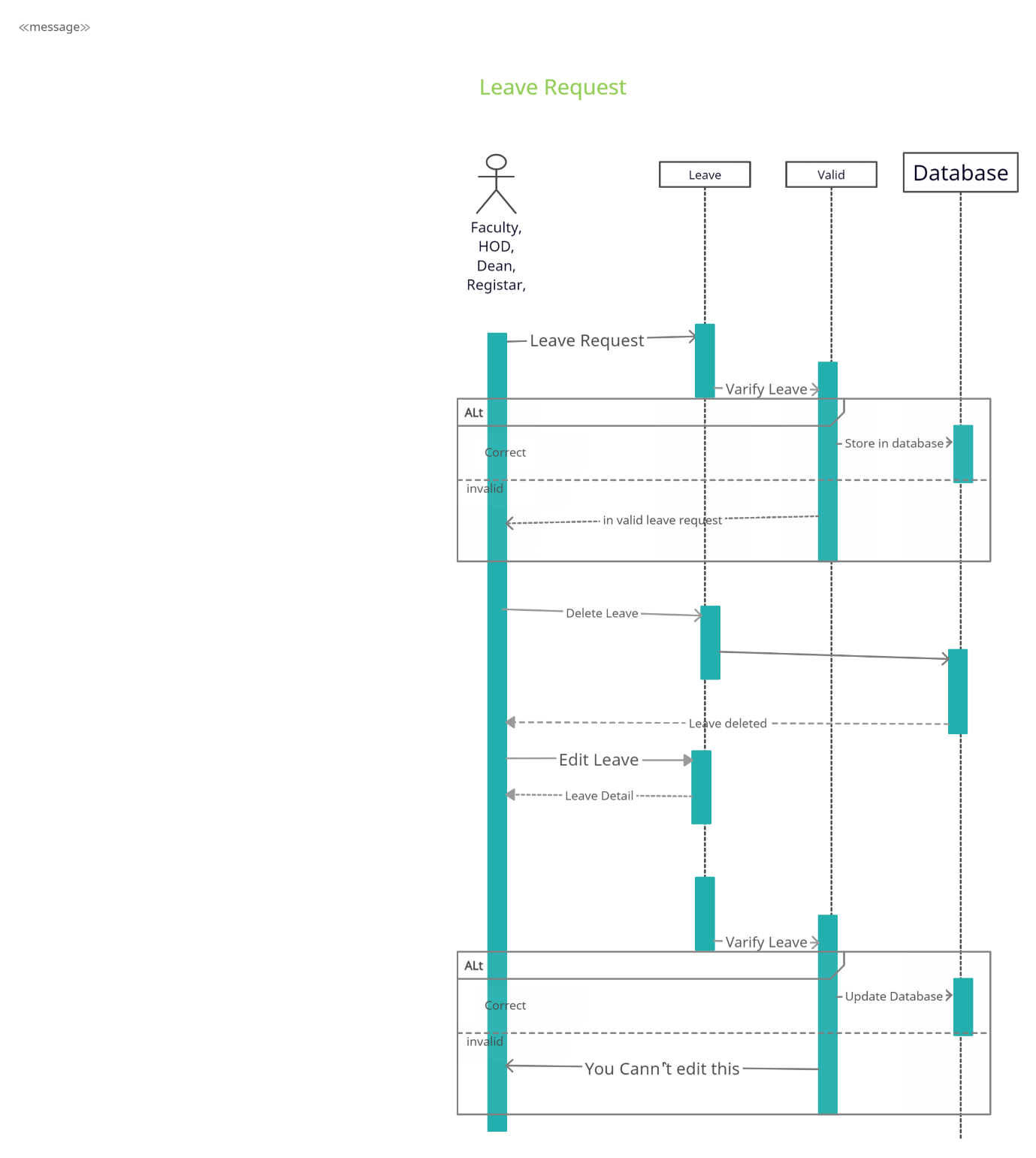
**3.1. Use-Case Diagram:**

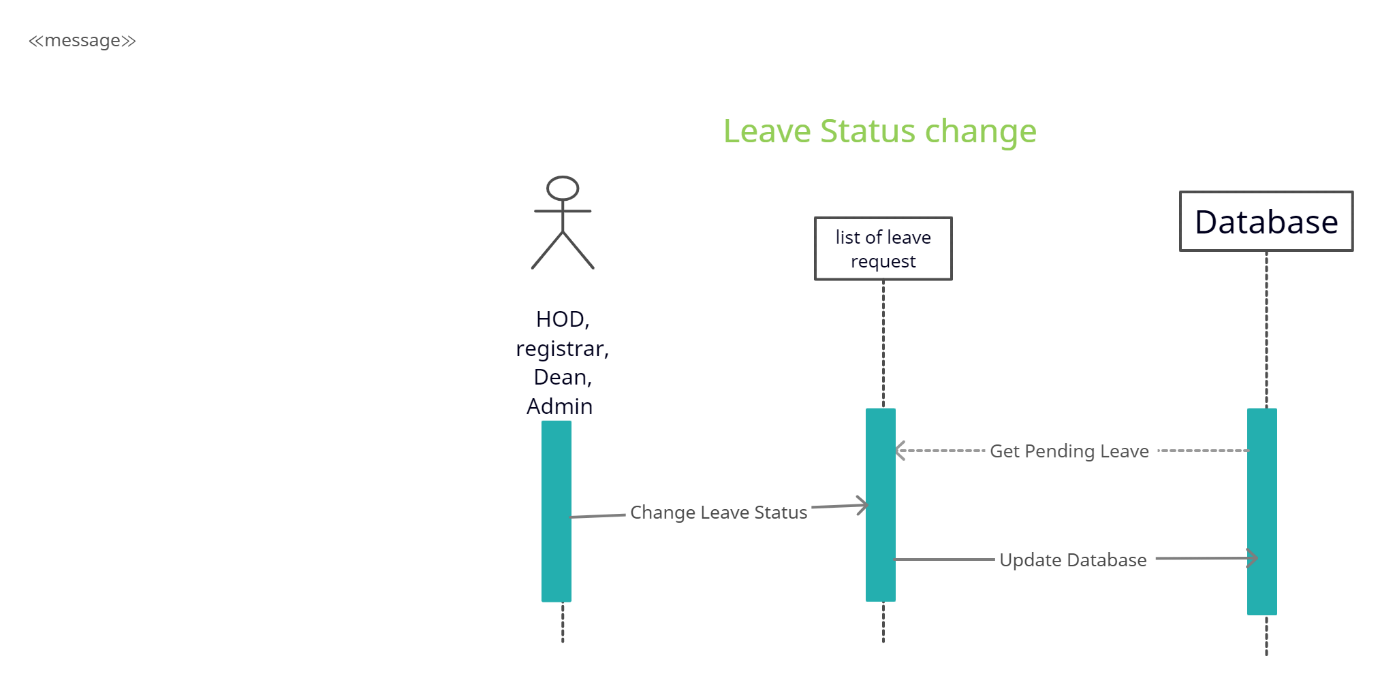
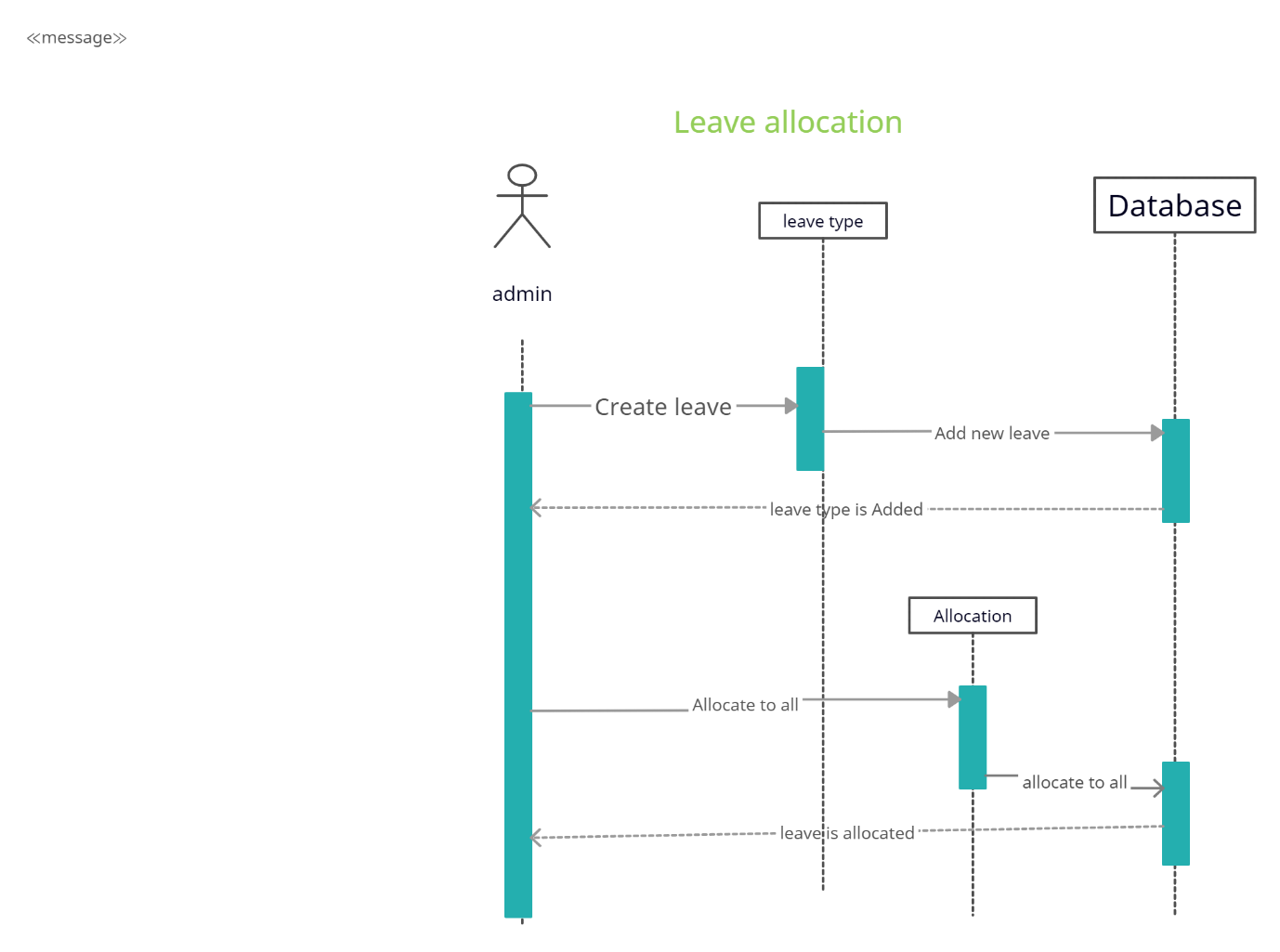


**3.2. Class Diagram**



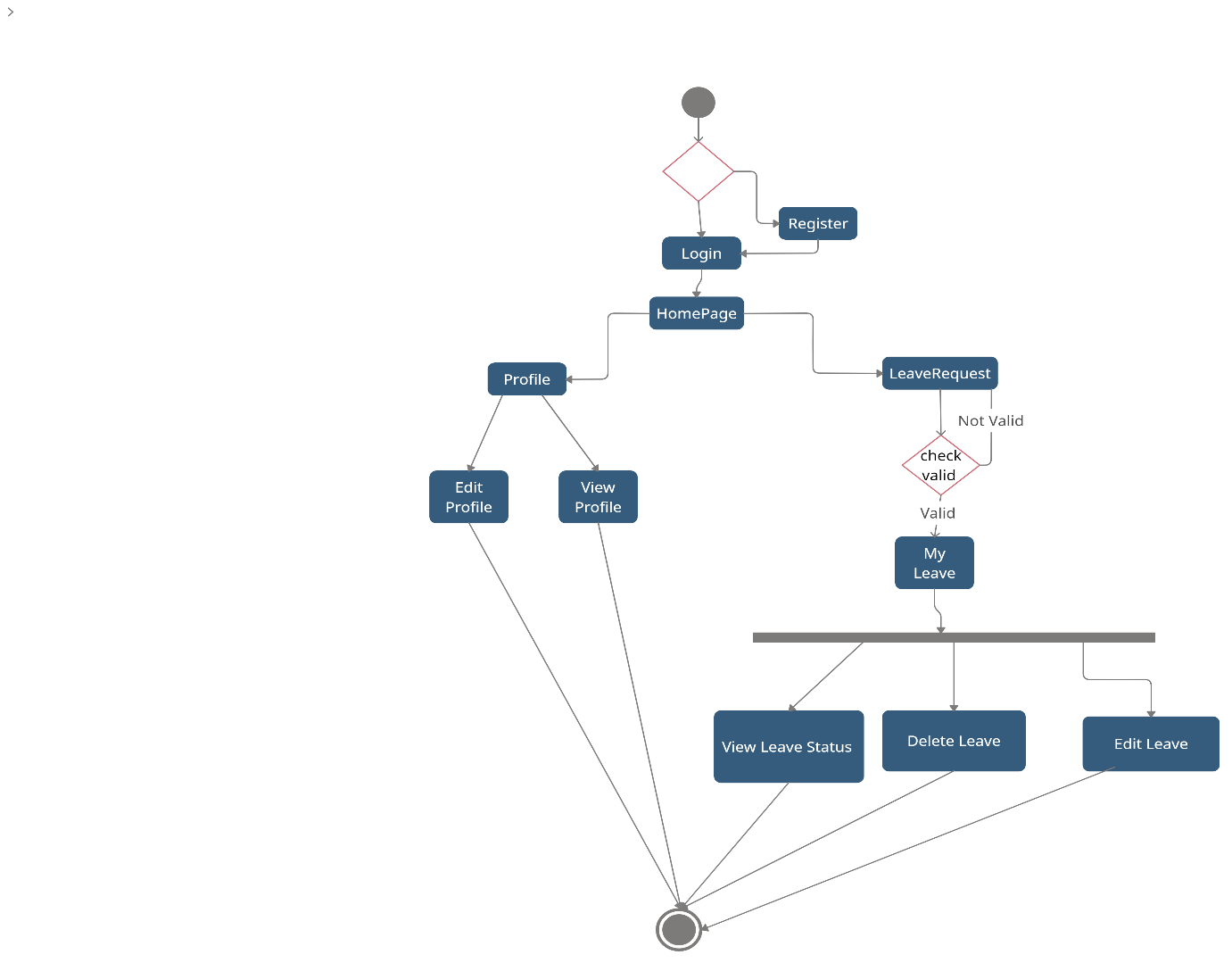
**3.3. Sequence Diagram:**



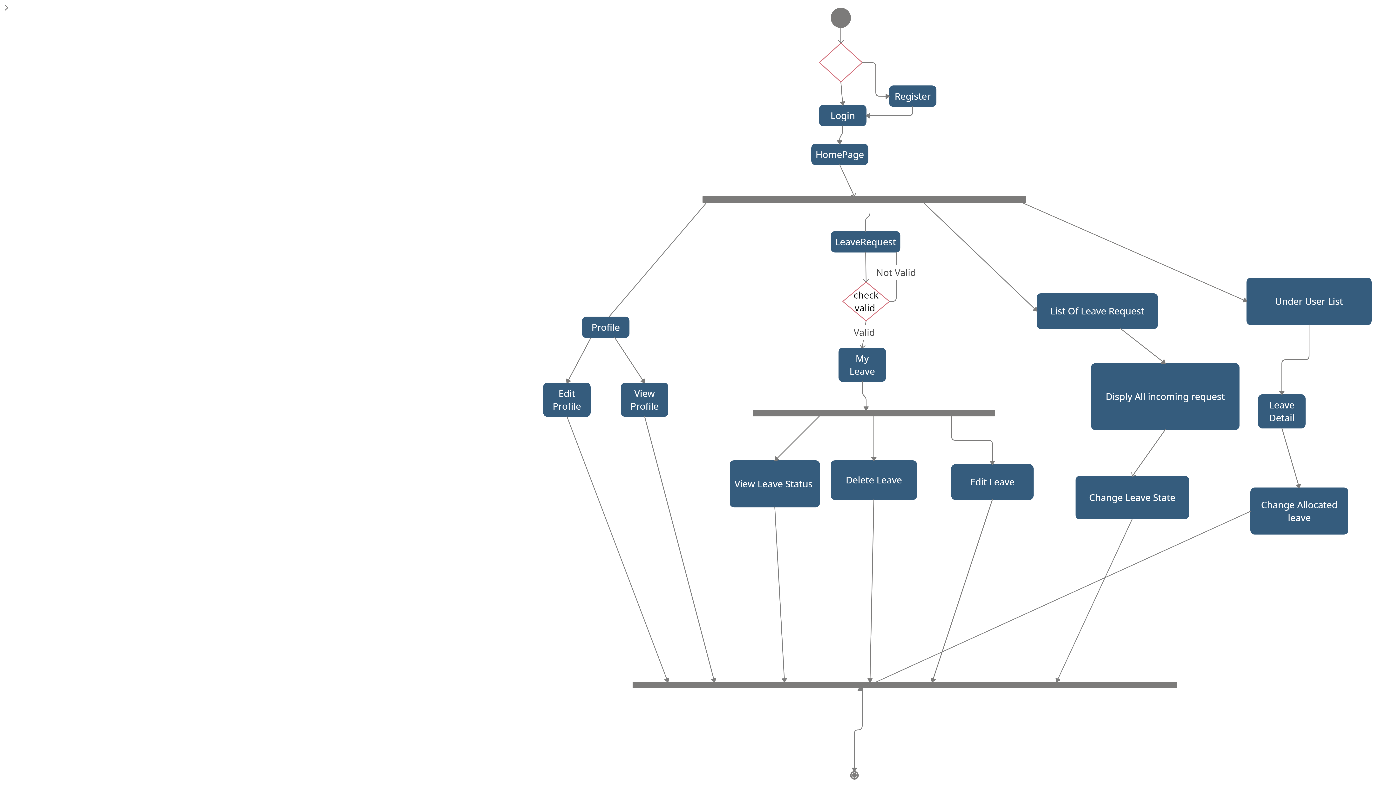


**3.4. State Diagram**

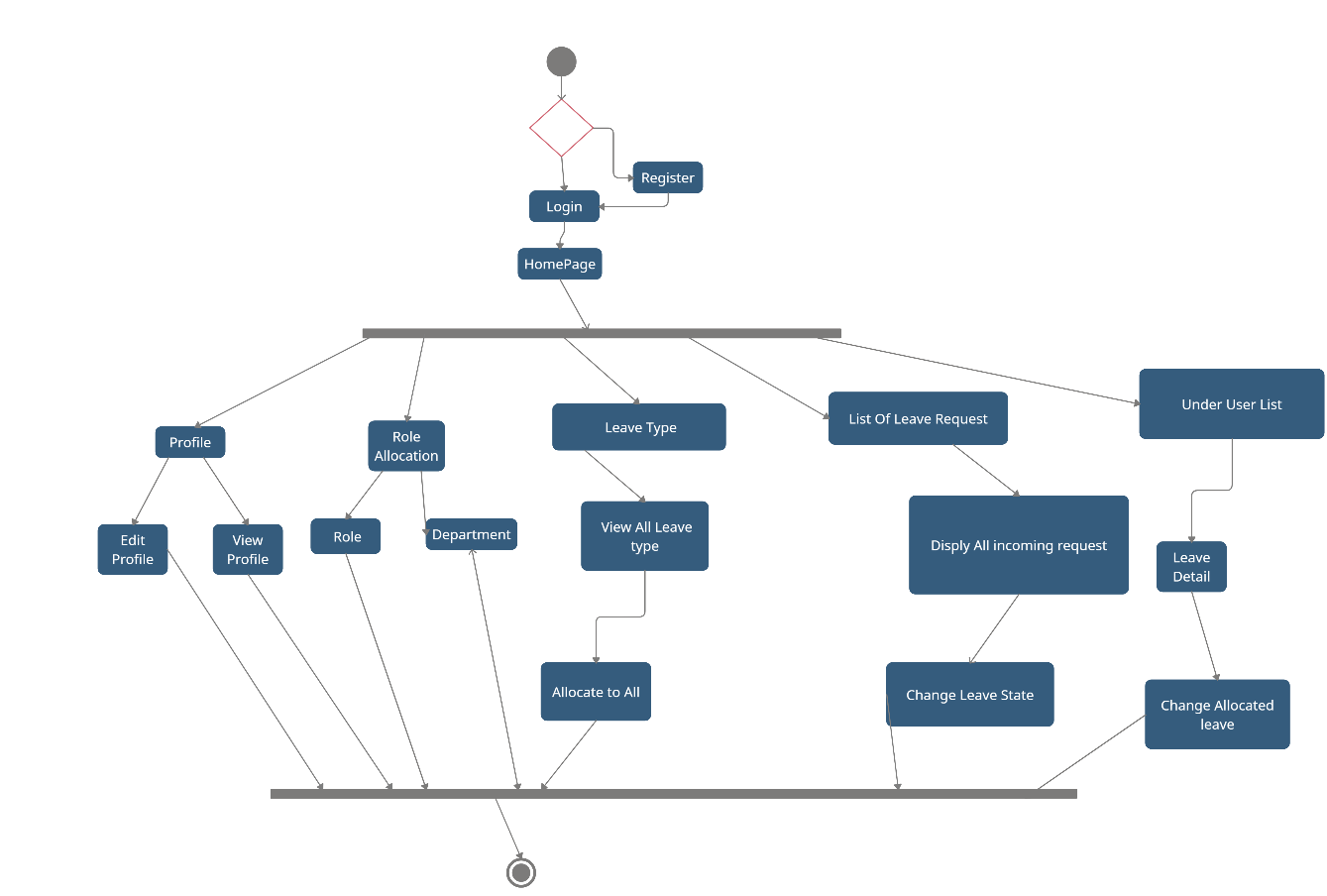
**3.4.1 Faculty**



**3.4.2 HOD Dean Register**

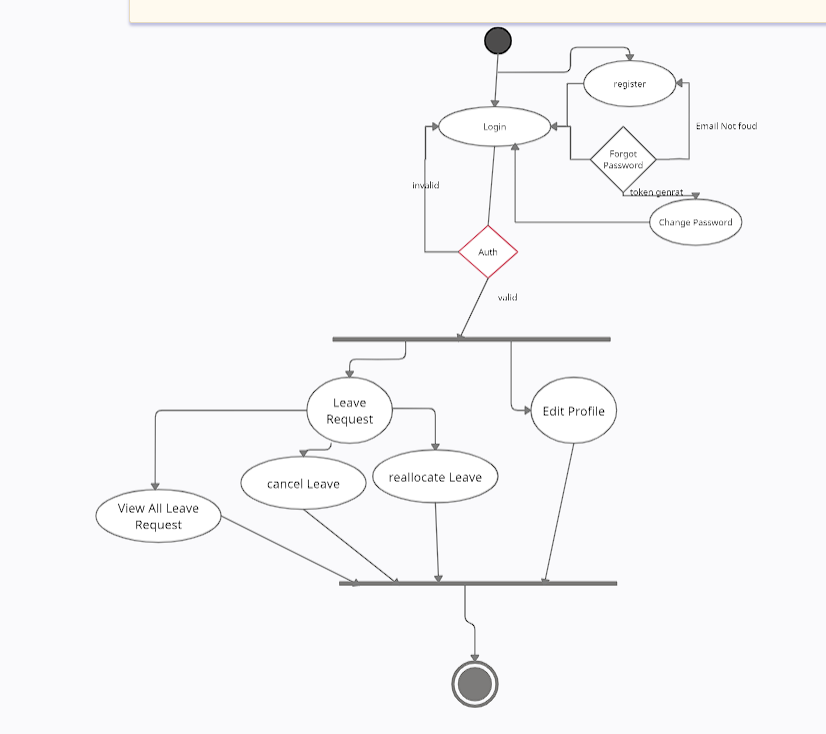


**3.4.3 Admin**

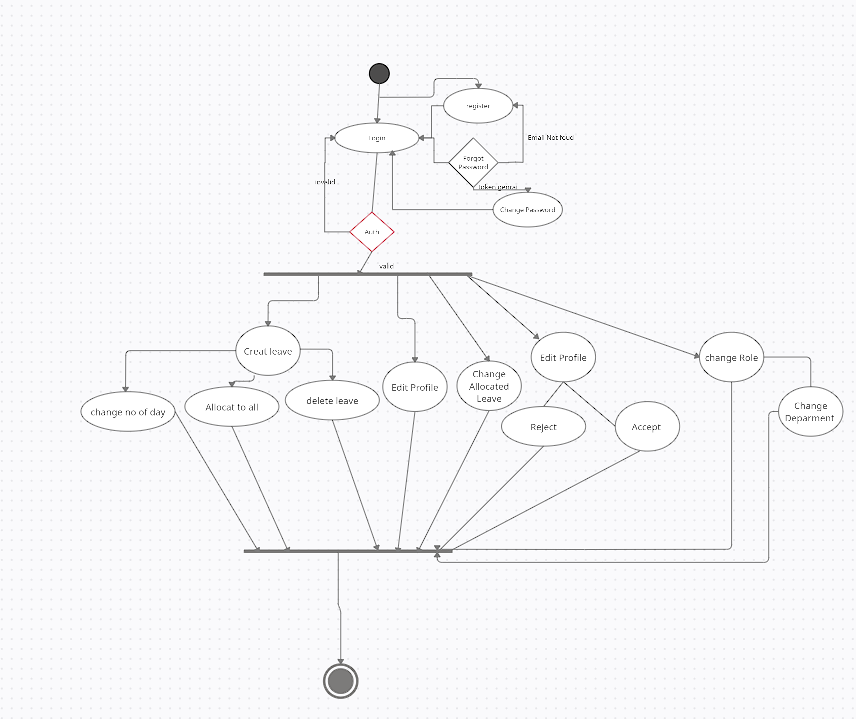


**3.5. Activity Diagram**

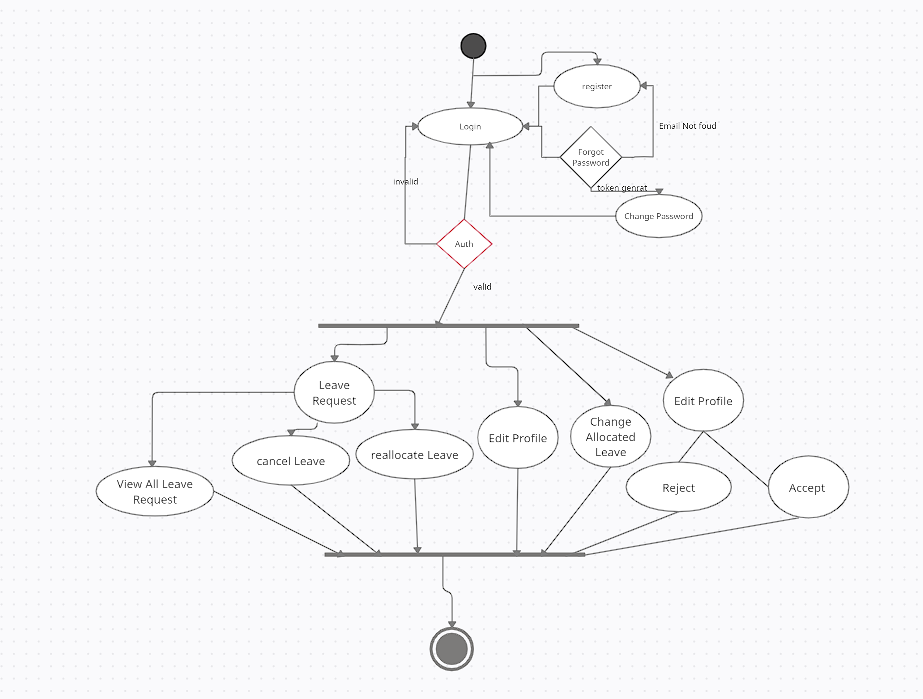
**3.5.1 Faculty**

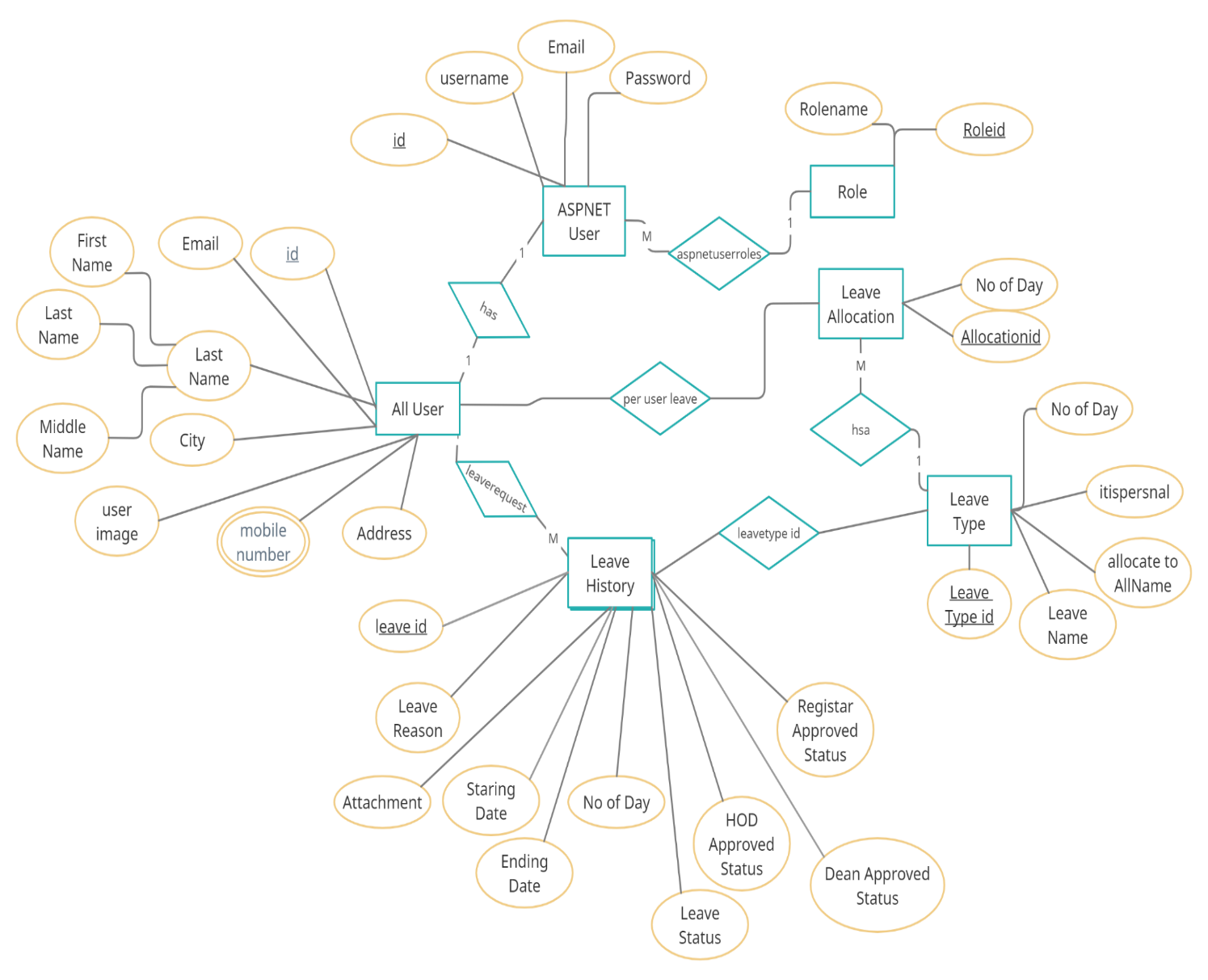


**3.5.2 Admin**



**3.5.3 HOD Dean Registrar**



**3.6. E-R Diagram**

**3.7. Data Dictionary**

The system Database consists of the following Table Schema:

**1. dbo.AllUser**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data Type** | **Size** | **Constraint** | **Reference** |
| Id | int |  | PK |  |
| Email | nvarchar | (MAX) |  |  |
| MobileNo | nvarchar | (MAX) |  |  |
| MobileNo2 | nvarchar | (MAX) |  |  |
| Name | nvarchar | (MAX) |  |  |
| LastName | nvarchar | (MAX) |  |  |
| MiddleName | nvarchar | (MAX) |  |  |
| Department | nvarchar | (MAX) |  |  |
| Role | nvarchar | (MAX) | , |  |
| Addreaddress | nvarchar | (MAX) |  |  |
| City | nvarchar | (MAX) |  |  |
| UserImage | nvarchar | (MAX) |  |  |

**2. dbo.** **AspNetRoles**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data Type** | **Size** | **Constraint** | **Reference** |
| Id | nvarchar | 450 | PK |  |
| Name | nvarchar | 256 |  |  |
| NormalizedName | nvarchar | 256 |  |  |
| ConcurrencyStamp | nvarchar | MAX |  |  |

**3. dbo.** **AspNetUserRoles**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data Type** | **Size** | **Constraint** | **Reference** |
| UserId | nvarchar | 450 | PK,FK | dbo.  AspNetRoles |
| RoleId | nvarchar | 450 | FK | dbo.  AspNetRoles |

**4. dbo.** **leaveAllocation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data Type** | **Size** | **Constraint** | **Reference** |
| AllocationID | int |  | PK |  |
| NoOfLeave | int | MAX |  |  |
| id | int | MAX | FK | dbo.AlllUser.id |
| leaveTypeID | int | 7 | FK | dbo.leaveType |

**5. dbo.** **AspNetUserTokens**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data Type** | **Size** | **Constraint** | **Reference** |
| UserId | nvarchar | 450 | PK,FK | dbo.  AspNetUsers |
| LoginProvider | nvarchar | 450 |  |  |
| Name | nvarchar | 450 |  |  |
| Value | nvarchar | MAX |  |  |

**6. dbo.** **LeaveHistory**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data Type** | **Size** | **Constraint** | **Reference** |
| leave\_id | int |  | PK |  |
| DeanApproveStatus | nvarchar | MAX |  |  |
| EndTill | datetime2 | 7 |  |  |
| HODApproveStatus | int | MAX |  |  |
| LeaveReason | nvarchar | MAX |  |  |
| LeaveStatus | nvarchar | MAX |  |  |
| NoOfDay | int |  |  |  |
| RegistrarApproveStatus | nvarchar | MAX |  |  |
| StartFrome | Datetime2 | 7 |  |  |
| id | Int |  | FK | dbo.AllUser |
| Attachment | nvarchar | MAX |  |  |
| leaveTypeID | int |  | FK | dbo.LeaveType |
| approved\_id | int |  |  |  |

**7. dbo.** **leaveType**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data Type** | **Size** | **Constraint** | **Reference** |
| leaveTypeID | int |  | PK |  |
| LeaveType | nvarchar | MAX |  |  |
| allcatoToAll | bit |  |  |  |
| itispersonal | bit |  |  |  |
| noofday | int |  |  |  |

**Chapter 4 - Implementation Details**

**Brief Description of Modules**

* The system consists of 17 basic modules namely

1. All User Module
2. Leave Allocation Module
3. Leave History Module
4. Leave Type Module
5. Leave Request Module
6. My Leave Module.
7. Status Module
8. Delete Leave Module
9. Edit Image Module
10. Forgot Password Module
11. Leave Request Module
12. Login Module
13. Register Module
14. Role Creation Module
15. Role Allocation Module
16. SMTP Module
17. Upload PDF Module

Each module consists of several methods to implement the required functionality. Implementation is done using. **Net5.0 Core MVC**. Database used in these modules is **MSSQLLocalDB**.

**Chapter 5 - Test case Design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model Name** | **Description** | **Expected Output** | **Actual output** | **Success or Fail** |
| Login | All data are correct | Redirect to home page | Redirect to home page | Success |
| Login | If credential are not correct | Invalid login attempt | Invalid login attempt | Success |
| Register | Everything is fine | Redirect to Login page | Redirect to Login page | Success |
| Register | -User already exist  -Password is not complex as over requirement  - conform password and password are not same | -Show appropriate message  -stay in same page | Show appropriate message  - stay in same page | Success |
| Leave Request | Everything is fine | Leave request done  Redirect to my leave page | Leave request done  Redirect to my leave page | Success |
| Leave Request | -If required field is not fill by user  -staring date > today`s date  - staring date < end till date | -Show appropriate message  -stay in same page | -Show appropriate message  -stay in same page | Success |
| Edit profile | Everything is fine | -profile is updated  -Redirect to home page | -profile is updated  -Redirect to home page | Success |
| Edit profile | -If required field is not fill by user | -Show appropriate message  -stay in same page | -Show appropriate message  -stay in same page | Success |
| Forgot Password | Email sent link is valid and password validation is done | -Redirect to login page  -password is also change | -Redirect to login page  -password is also change | Success |
| Forgot Password | Email sent link is not valid and password validation is not done | -Show appropriate message  -stay in same page | -Show appropriate message  -stay in same page | Success |

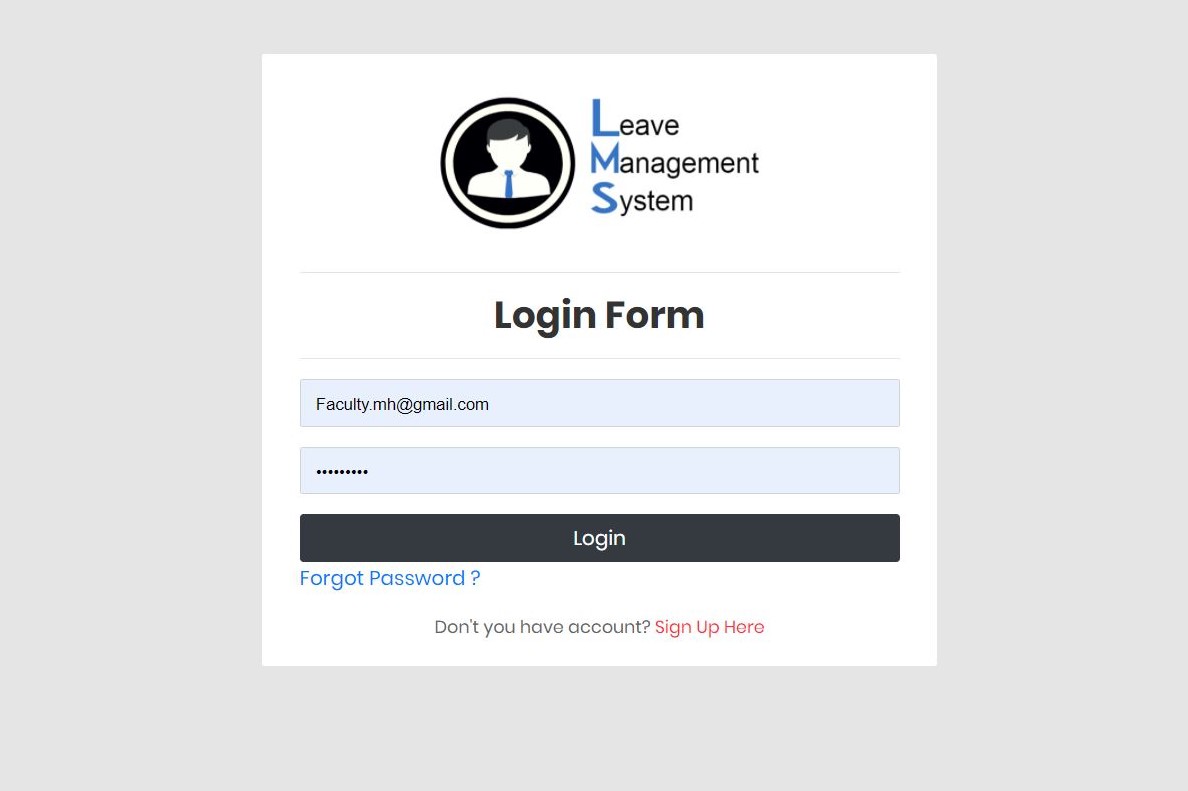
**Chapter 6 – Screenshots**

Figure 1: Login Page

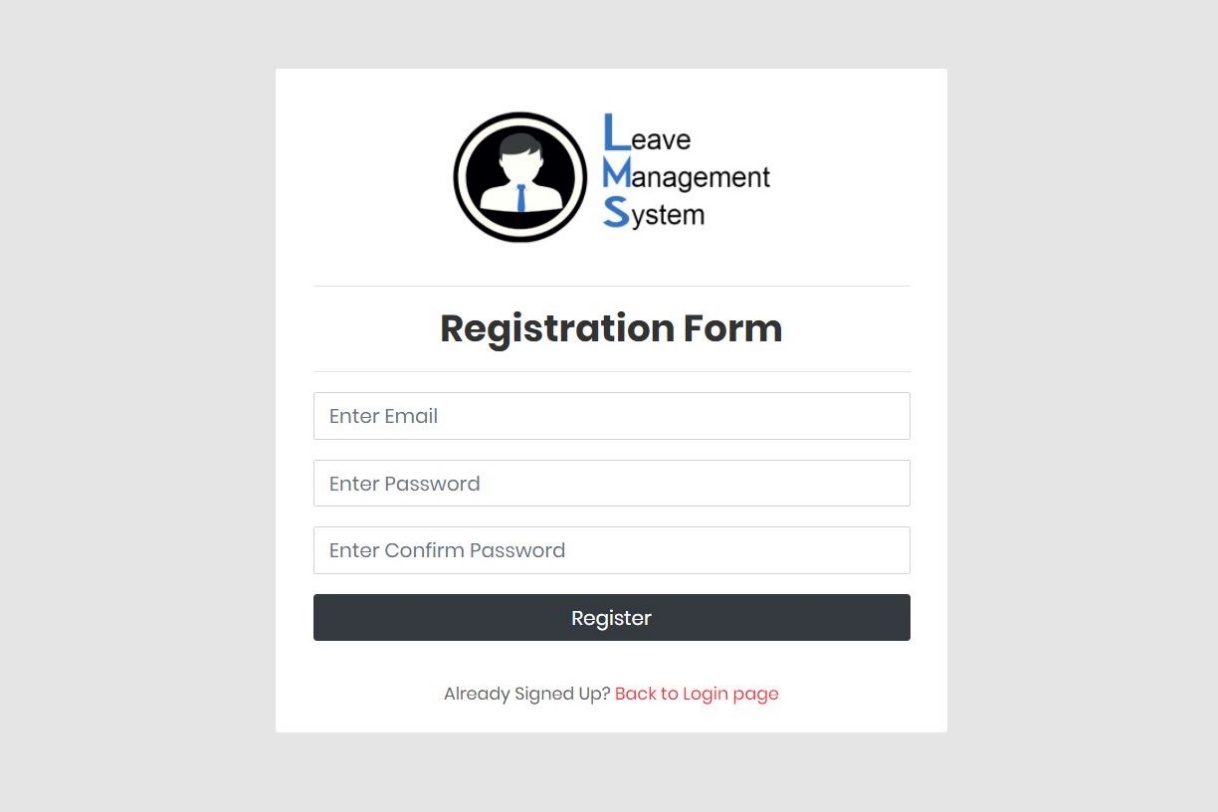
****

Figure 2: Registration Form

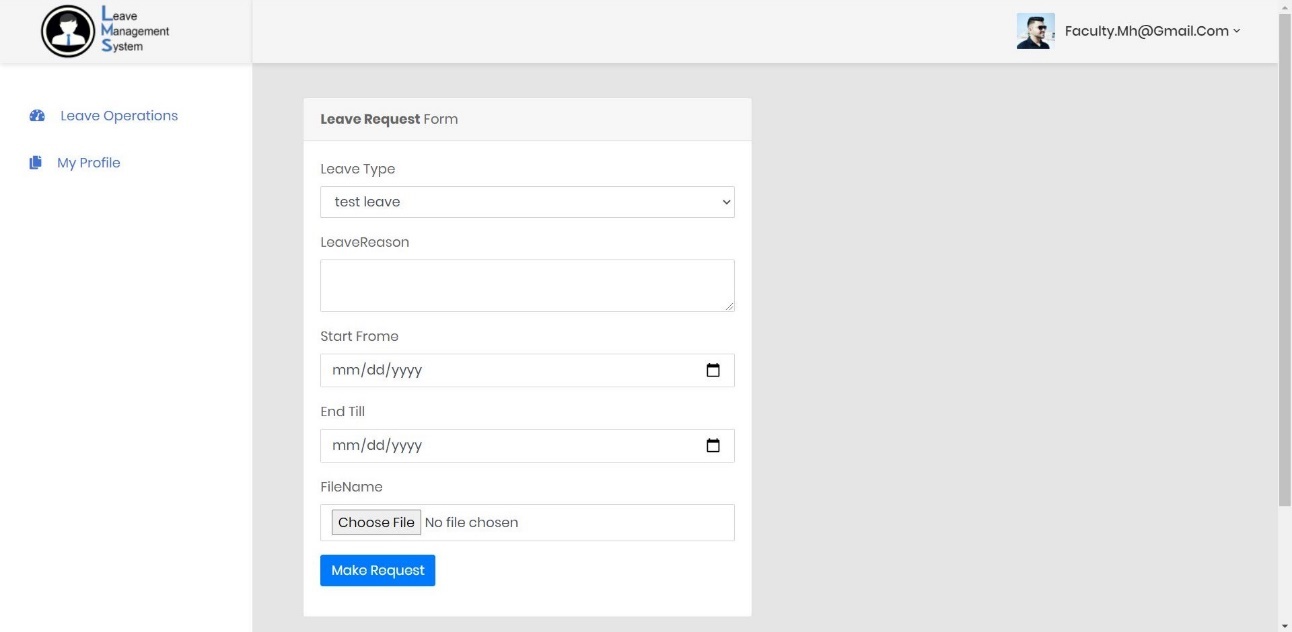


Figure 3: Create leave request Form Faculty Side.

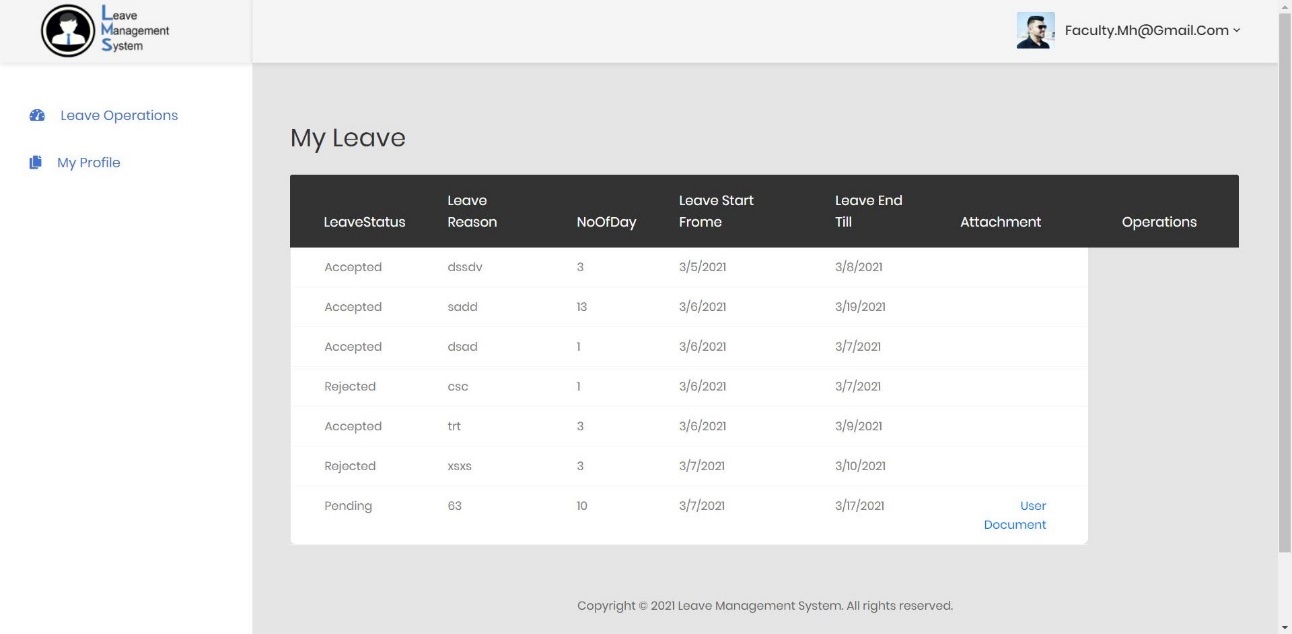


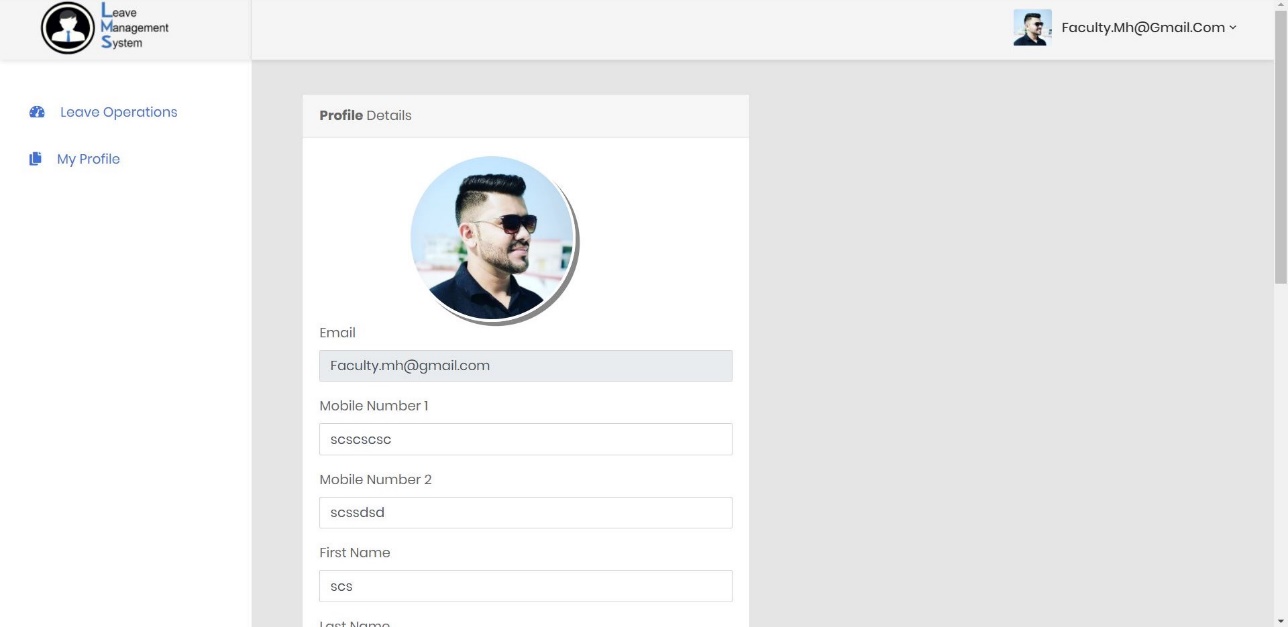
Figure 4: Requested Leave (My Leave) Page Faculty side

Figure 5: Profile Details Page Faculty Side

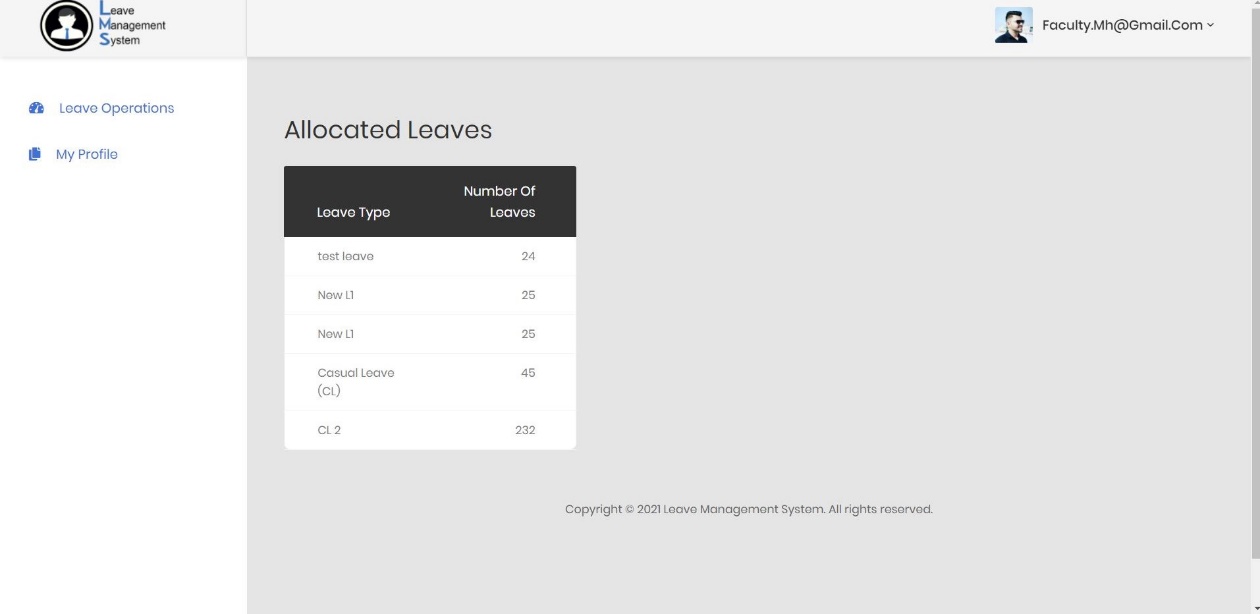


Figure 6: List of available Leave Faculty Side

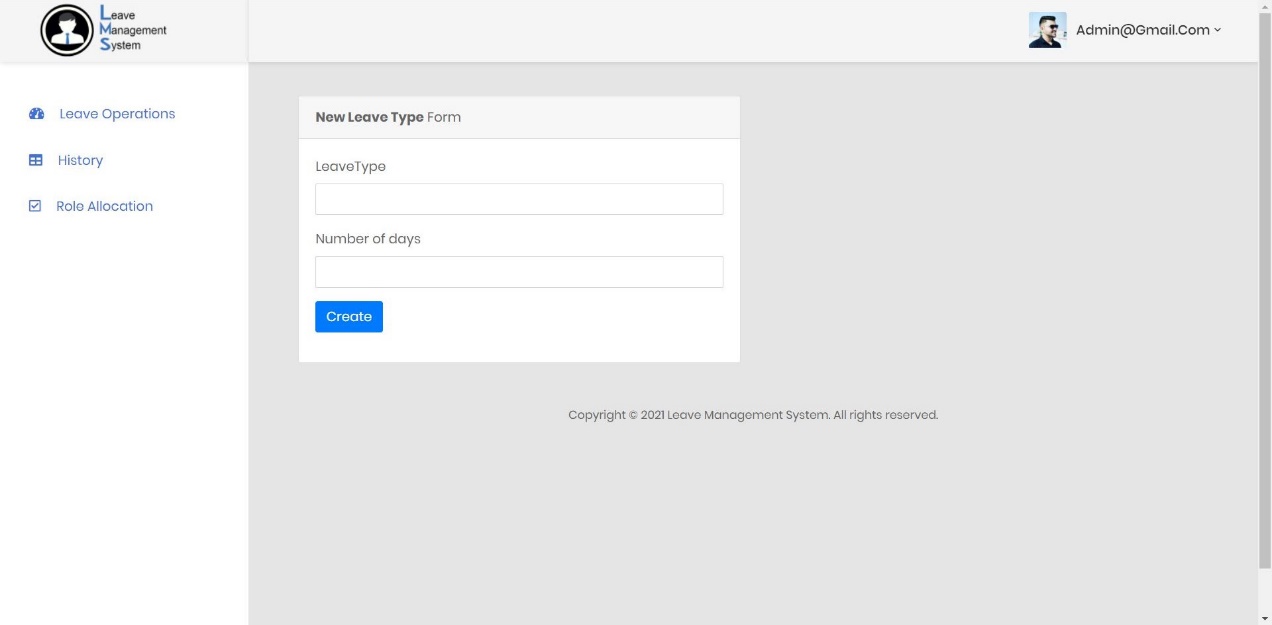


Figure 7: Leave Type Create Page Admin Side

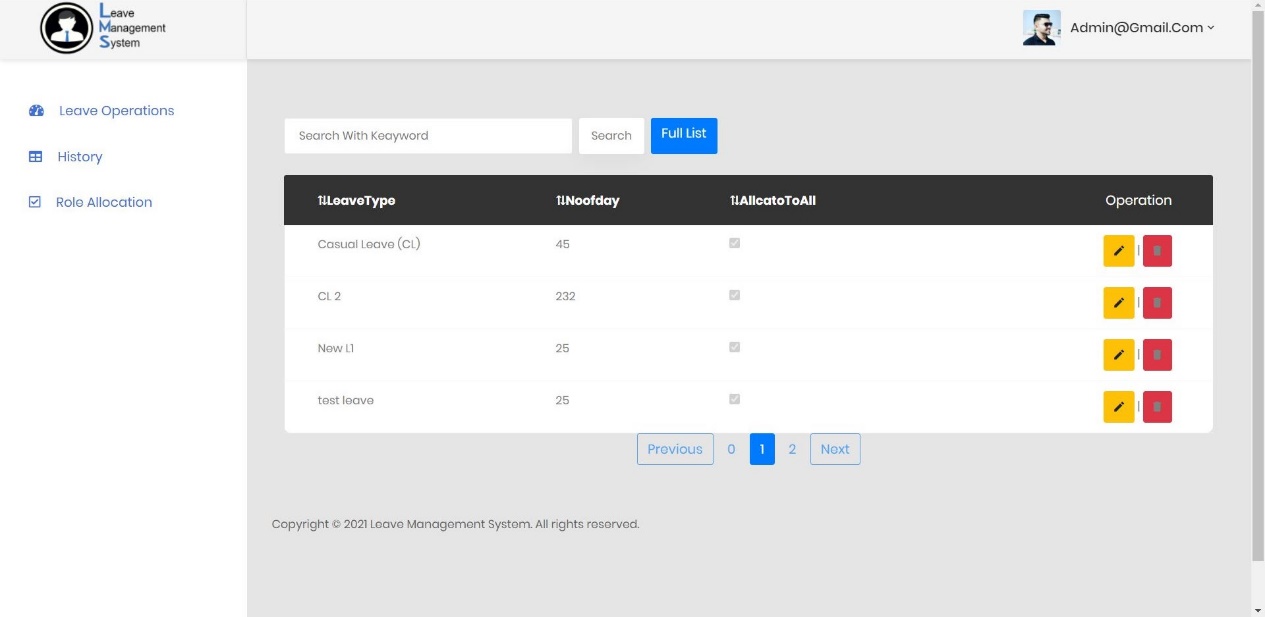


Figure 8: List of leave type Admin Side

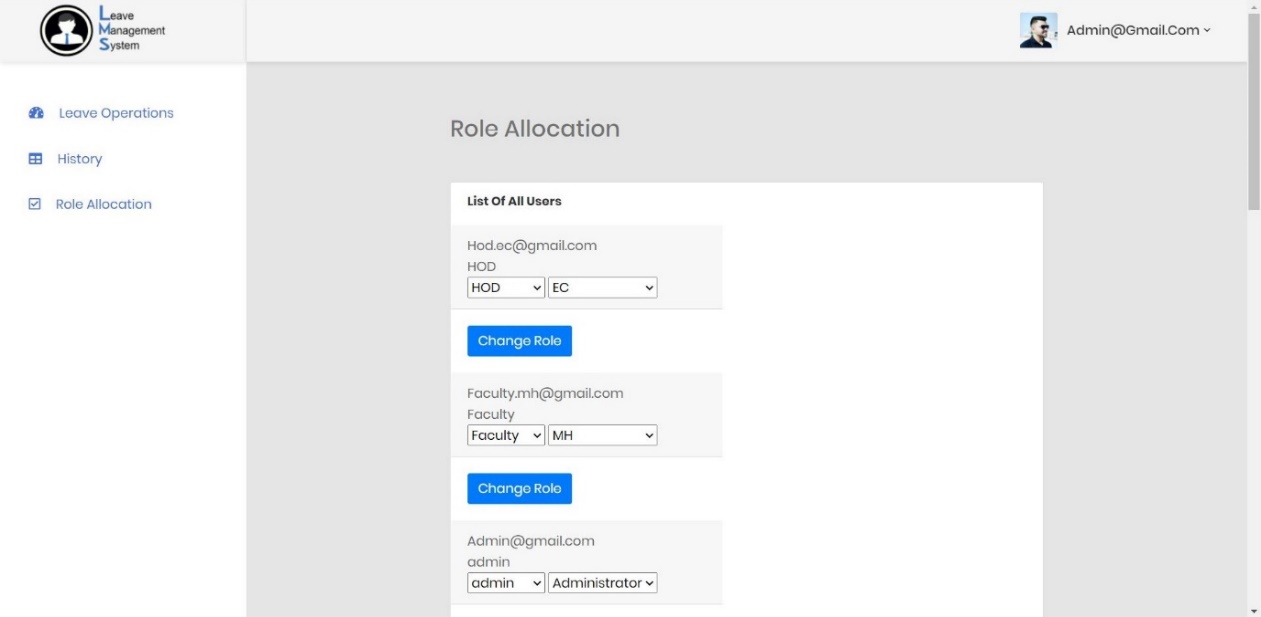


Figure 9 : Role Allocation Module Admin Side

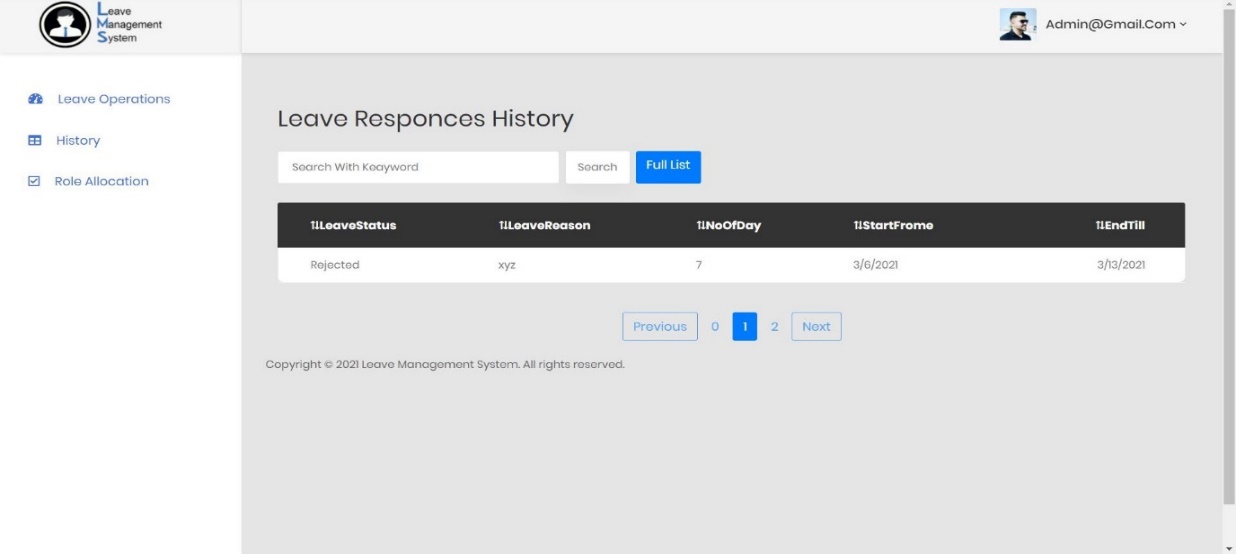


Figure 10 : Leave Responses history Admin Side

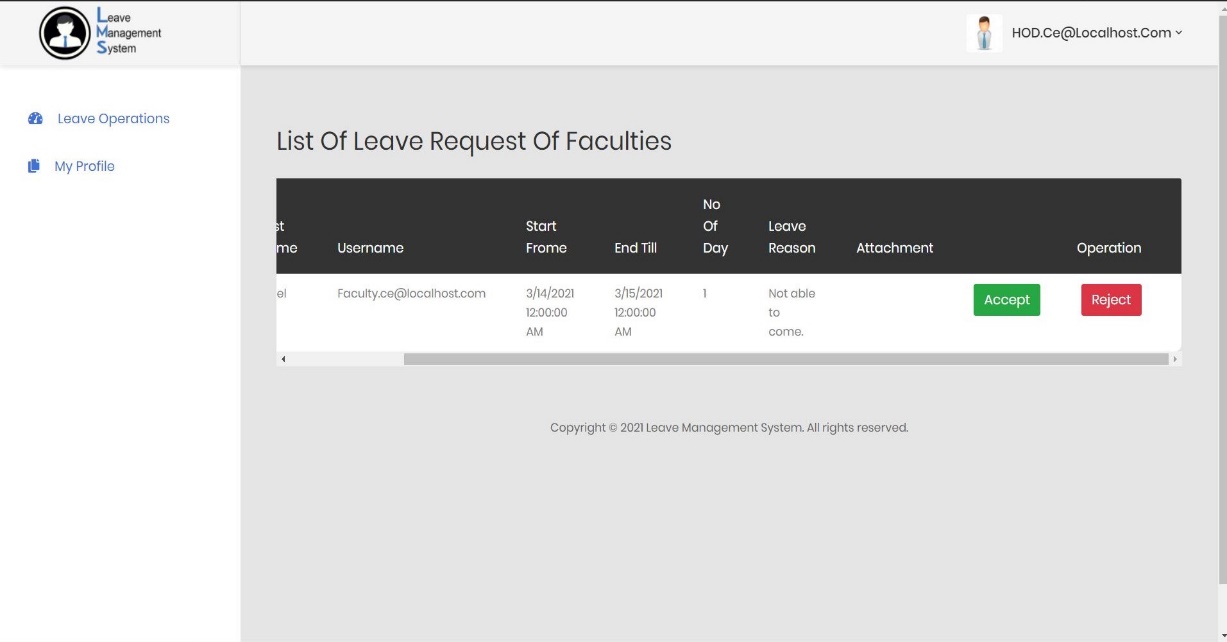


Figure 11: Pending Leave Requests page

**Chapter 7 - Conclusion and further extensions**

**7.1 Conclusion:**

The functionalities are implemented in system after understanding all the system modules according to the requirements. Functionalities that are successfully implemented in the system are:

* **User Sign Up**
* **User Sign In (Pending user, Faculty, HOD, Dean, Registrar, Admin)**
* **User Profile Details Updating.**
* **Create New Leave Type**
* **Leave Allocation**
* **Create Leave Requests (Faculty, HOD, Dean, Registrar)**
* **Attachment Upload in Leave Request.**
* **Respond to Leave Requests (HOD, Dean, Registrar, Admin)**
* **Edit Allocated Leaves (HOD, Dean, Registrar, Admin)**
* **List Available Remaining Leaves of their under users.( HOD, Dean, Registrar, Admin).**
* **All Leave History.**
* **All Leave Response history.**
* **Leave Updates sends in Email.**
* **Forgot password using sending Email**
* **Role Allocation**

After the implementation and coding of system, comprehensive testing was performed on the system to determine the errors and possible flaws in the system.

**7.2 Further Extension**

As this is a Leave Management platform the extensions are limitless Google Calendar is the latest technology which can be implemented in our system which will make product sync with google service and data retrieval will be very efficient and fast.

Lately we are have planned to integrate My Calendar which and lead our task very easy and manageable for all the users.

**Chapter 8 - Reference / Bibliography**

1. [**https://docs.microsoft.com/en-us/aspnet/core/data/ef-mvc/sort-filter-page?view=aspnetcore-5.0**](https://docs.microsoft.com/en-us/aspnet/core/data/ef-mvc/sort-filter-page?view=aspnetcore-5.0)
2. [**https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-5.0**](https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-5.0)
3. [**https://docs.microsoft.com/en-us/dotnet/api/system.net.mail.smtpclient?view=net-5.0**](https://docs.microsoft.com/en-us/dotnet/api/system.net.mail.smtpclient?view=net-5.0)
4. [**https://github.com/puikinsh/CoolAdmin**](https://github.com/puikinsh/CoolAdmin)
5. [**https://www.w3schools.com/css/**](https://www.w3schools.com/css/)
6. [**https://getbootstrap.com/**](https://getbootstrap.com/)
7. [**https://www.freecodecamp.org/news/asp-net-core-mvc-course-net-5/**](https://www.freecodecamp.org/news/asp-net-core-mvc-course-net-5/)
8. [**https://www.youtube.com/watch?v=P3shzPde8Eg**](https://www.youtube.com/watch?v=P3shzPde8Eg)