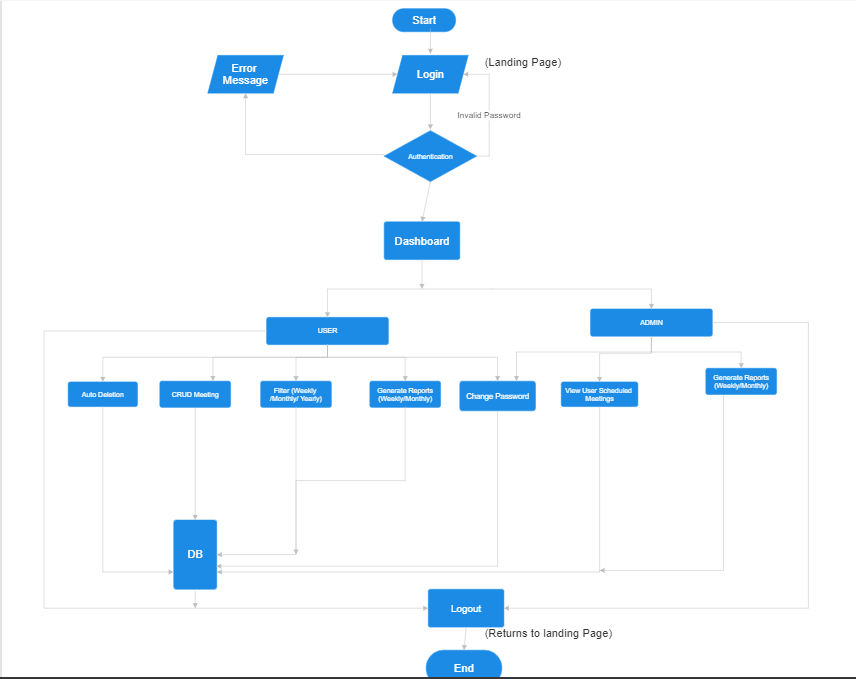
|  |  |
| --- | --- |
| **Description** | Meeting Scheduler Project. This is a web-based project and allows the users to log in to the web page. After logging in, the users can add, edit or even delete the meeting schedules. The feature of email sending should be a must in this project to remind the user of an upcoming meeting with its date and time |
| **Actors** | Users and Admins |
| **Functional requirements** | Logging in requires a user id and password. The user ID should be a valid email ID and the password is of the user’s choice. There should be authentication and security while Logging In and an option to change passwords should also be provided to the users. Support for Logout should be there |
|  | Meeting and User details data to be managed |
|  | Users can Log In and Log Out of the application. |
|  | Role management for Users and Admins, create separate views for Users and Admins |
|  | Admins can see every user's meetings in read-only mode |
|  | Users can only see their own meetings and can edit and delete those meetings |
|  | Users able to search scheduled meetings based-on Weekly, Monthly and Yearly |
|  | Needed reports for weekly and monthly meetings scheduled |
|  | Adding new meetings and deleting some of them. Auto-deletion of the meetings after the time has passed. |
|  |  |
| **Non-Functional requirements** | Microsoft suggested standards and patterns in coding and development |
|  | Industry standard application layering should be applied |
|  | Maximum extendibility and scalability to be maintained after development |

**Flow Chart:**



**Modules:**

1. Login module
2. User scheduler module
3. All Users scheduler module
4. Edit User module
5. Delete User module
6. Autodeletion module
7. Filters Module
8. EmailNotification Module
9. Change password Module
10. Reports Module

**Login module:**

* UI: Login page with two textboxes to get username and password, submit button.
* Logic: Validate the username and password.
* Authentication: Check the valid user
* Authorization: Check user role to access Admin/User page.
* Tables: tblUser, tblRole,
* tblUser -> Id, username, Password, roleId.
* tblRole-> Id, RoleName

**User Scheduler Module:**

* Frontend: Meeting creation form.
* Backend: Logic for creating meeting.
* Table: tblMeeting
* tblMeeting-> Id, Agenda, date, time, participant.

**Dashboard Module:**

* Frontend:

Create separate views for users and administrators, displaying relevant meeting details.

Admins should have access to all user meetings in read-only mode.

* Backend: Implementing to display meetings list based on Role.
* Tables: It brings data from tblRole & tblMeeting.

**Edit User Module:**

* Frontend: Editing form scheduled meeting.
* Backend: Logic for updating meeting details.
* Table: Updates the record in tblMeeting.

**Delete User Module:**

* Frontend: Meeting deletion form.
* Backend: Logic to delete scheduled meeting.
* Table: Delete record from tblMeeting.

**Auto-Deletion Module:**

* Backend: Automatically deletes the meeting based on time frame(past meetings).
* Table: Auto deletes the record from tblMeeting.

**Filter Module:**

* Frontend: Filter option for scheduled meetings.
* Backend: Implementing filter functionality.
* Table: Fetches data from tblMeeting(user-specific and admin(all users)).

**Email Notification Module:**

* Backend: Logic for sending E-mail Notification or remainder.
* Table:

**Change Password Module:**

* Frontend: Form for password changing.
* Backend: Logic for changing password.
* Table: Updates the password of a specific user in tblUser.

**Reports Module:**

* Frontend: Creating an interface for report generation based user choice(Weekly/Monthly).
* Backend: Logic to generate reports based on user choice.
* Table: