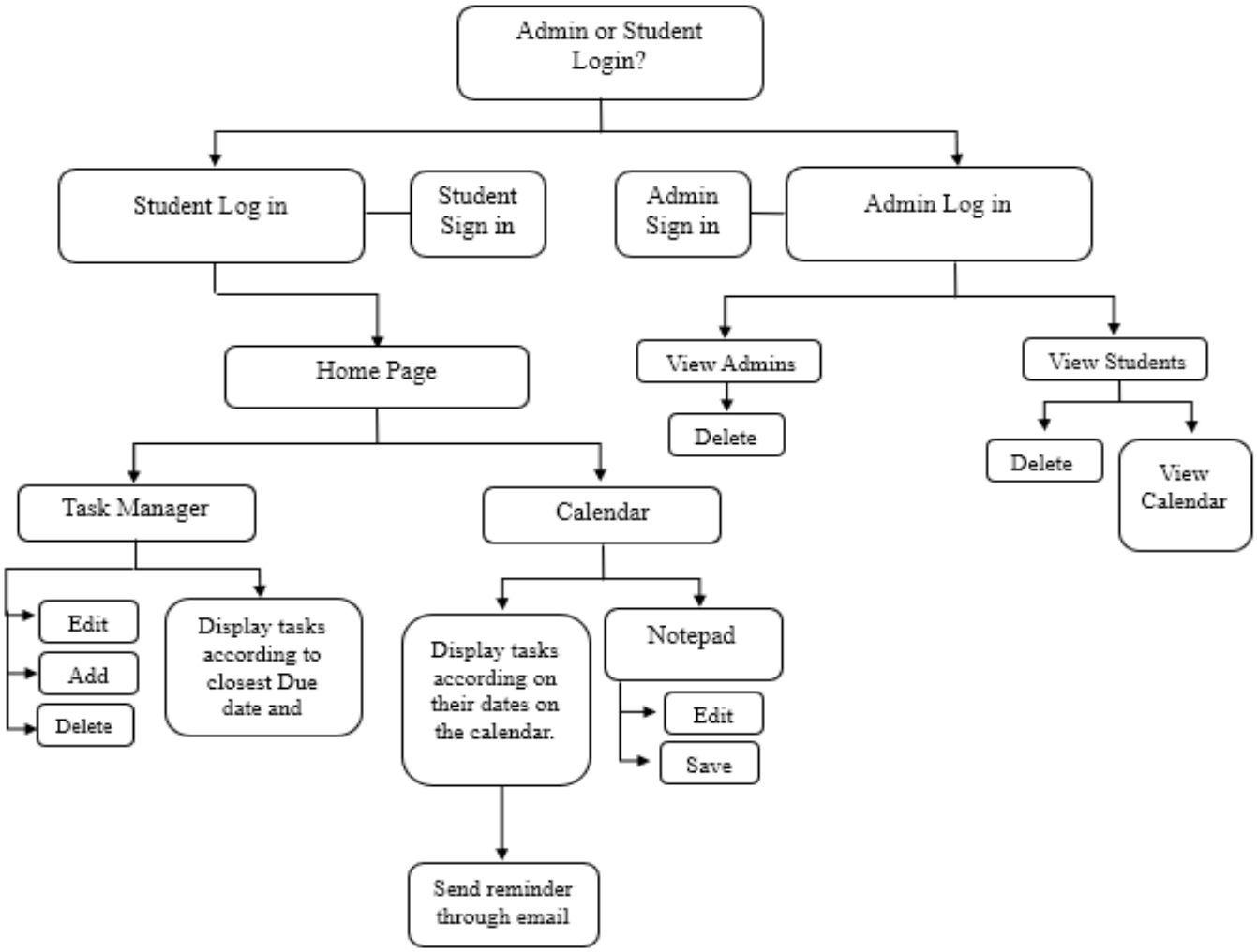
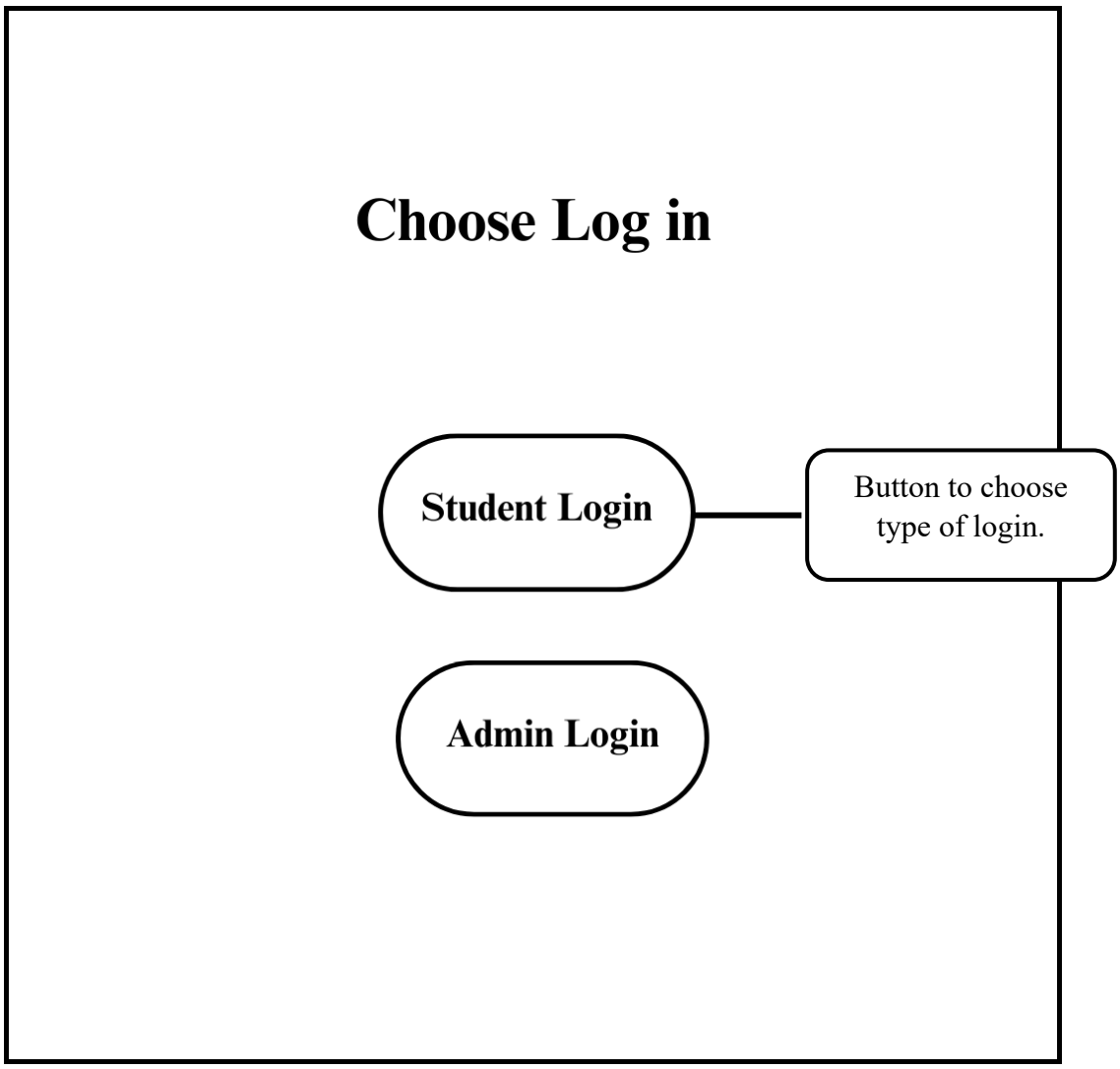
Criterion B

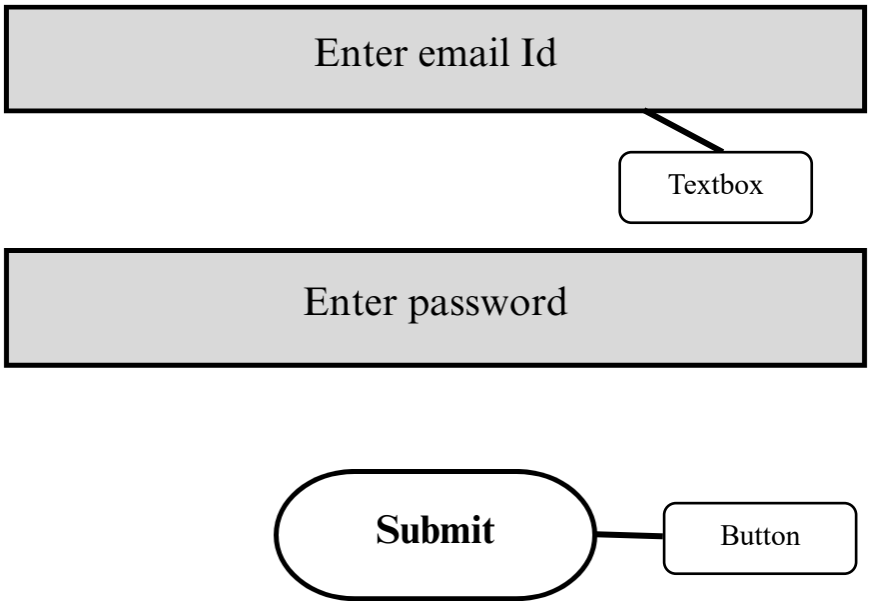
Application UI flow diagram



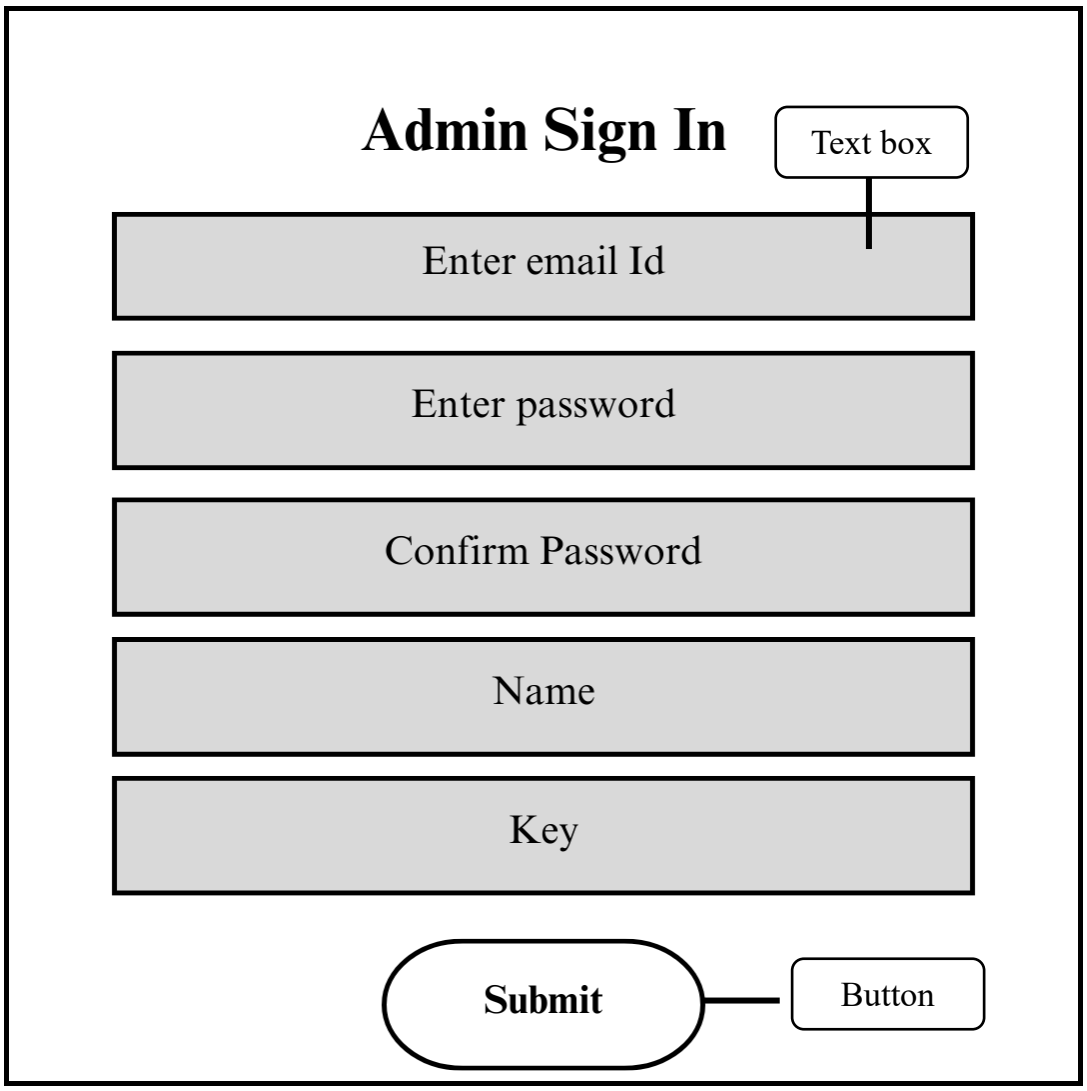
User Interface design

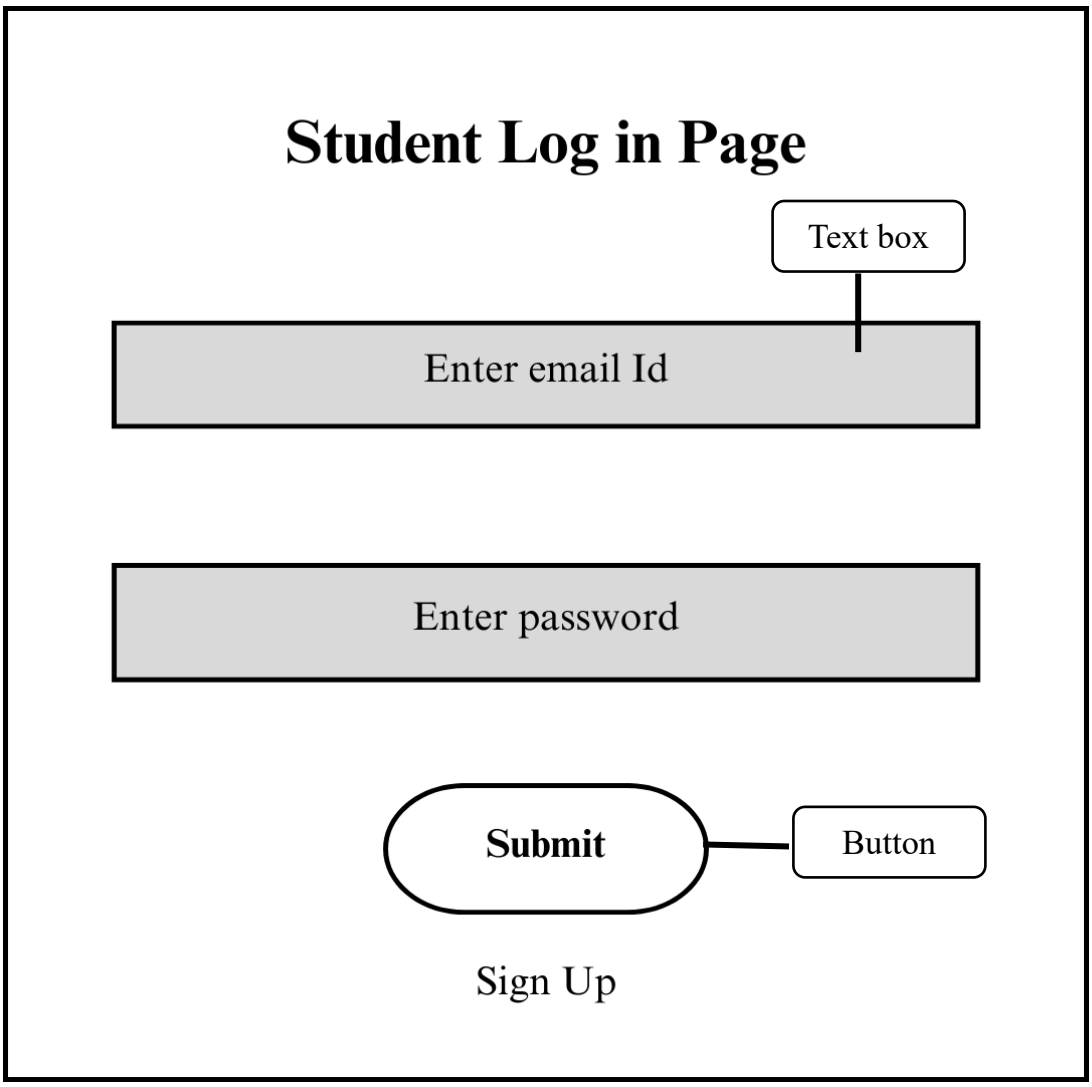


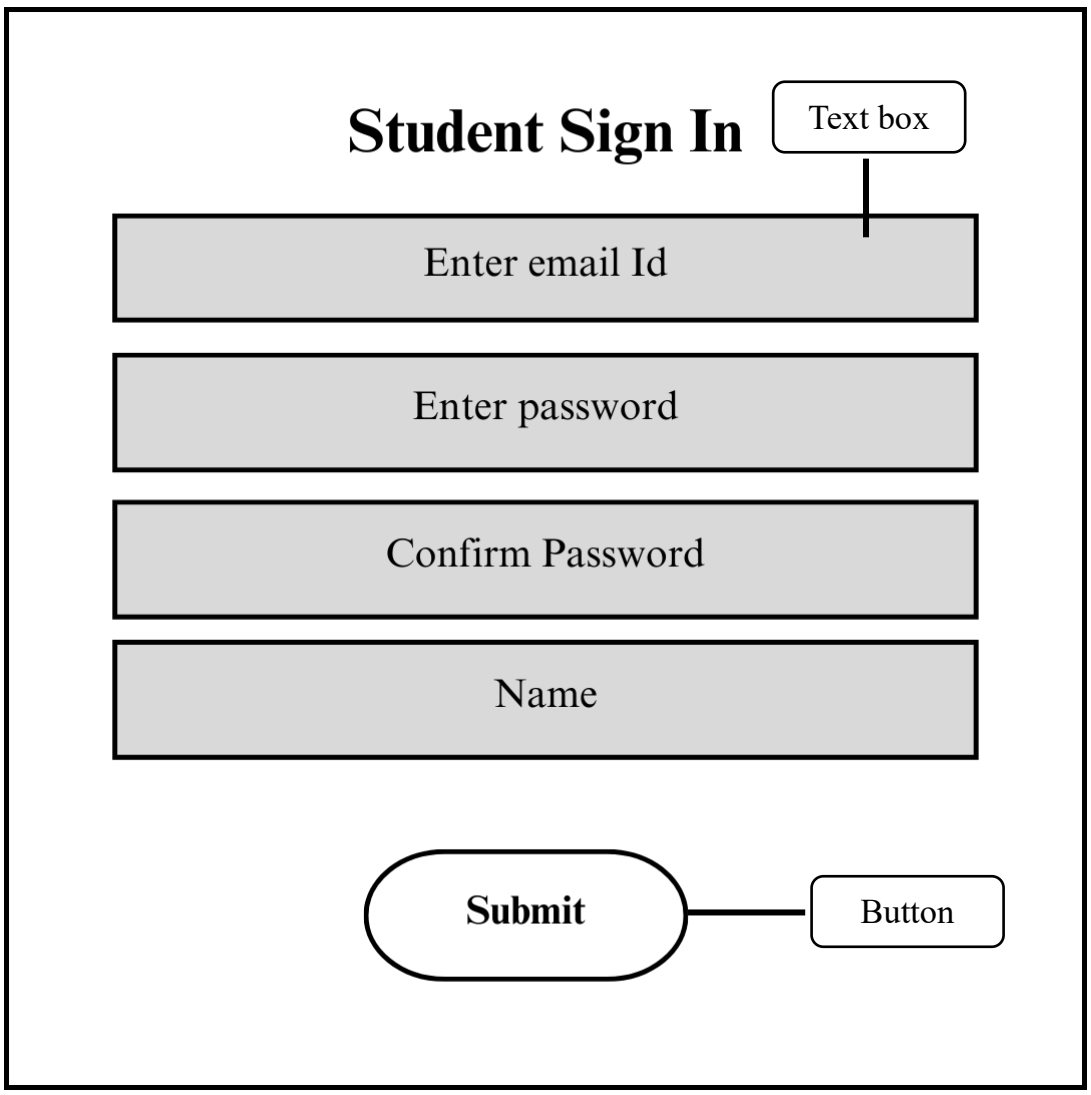
Admin Log in

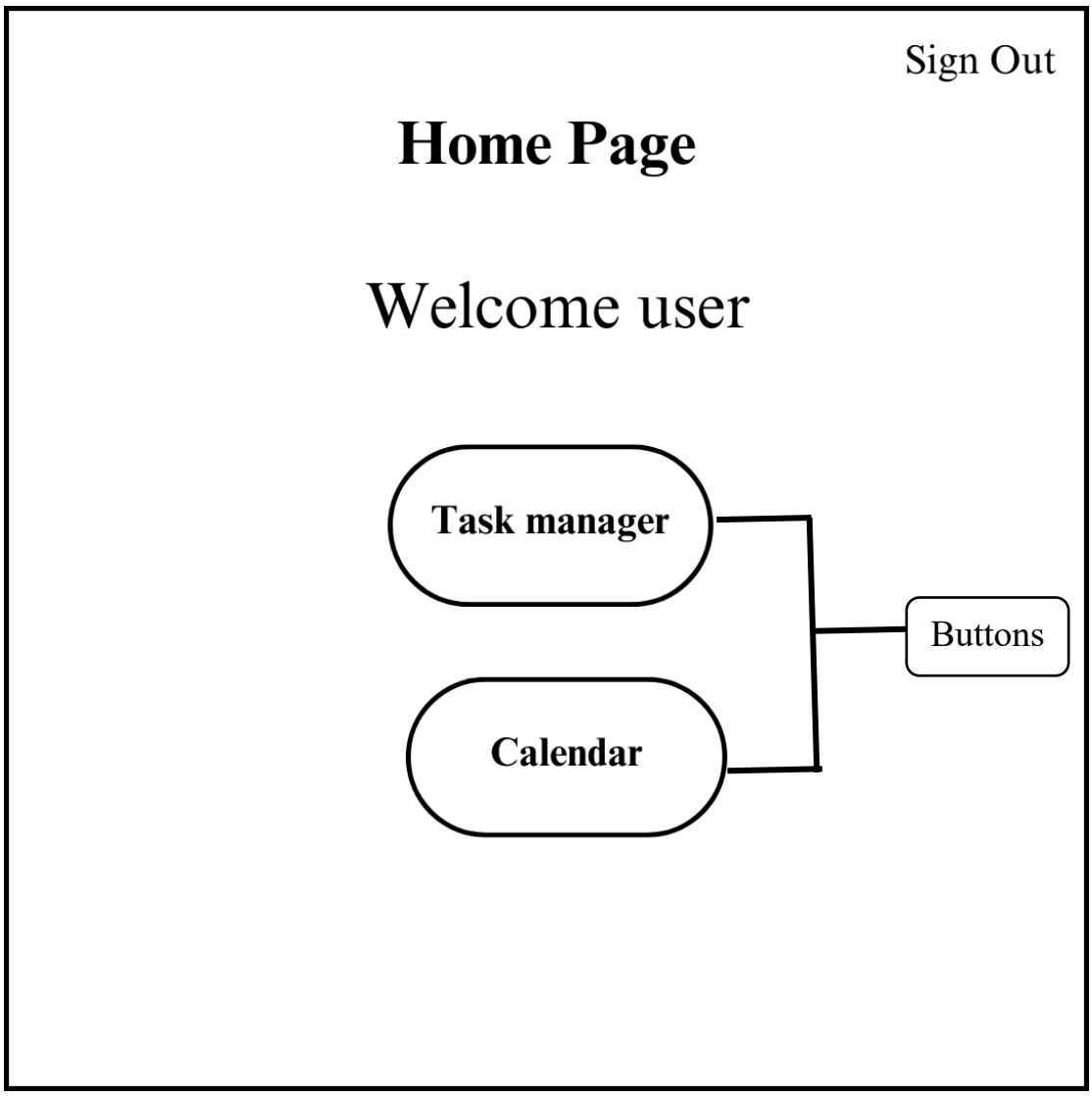


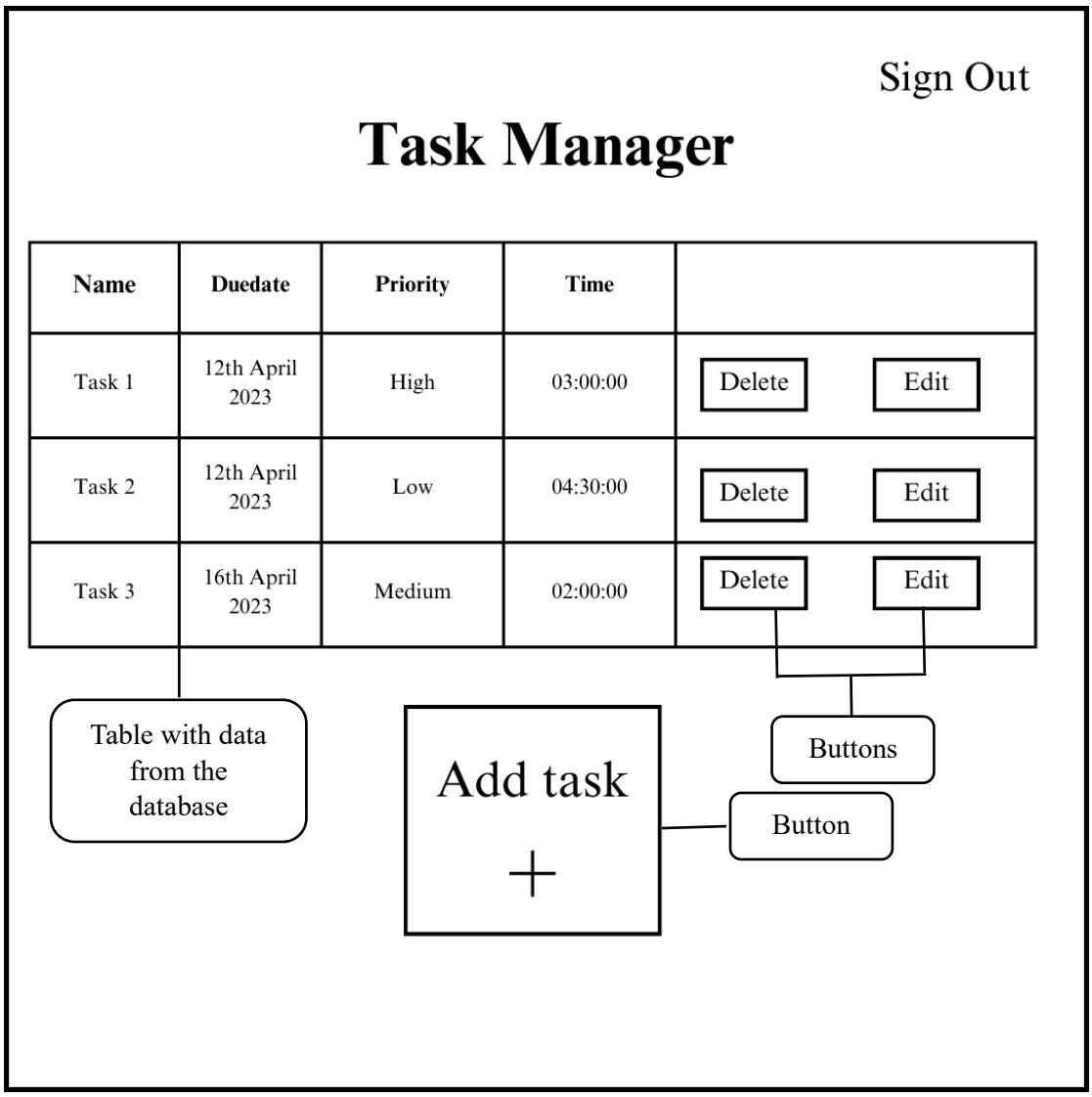
Sign Up

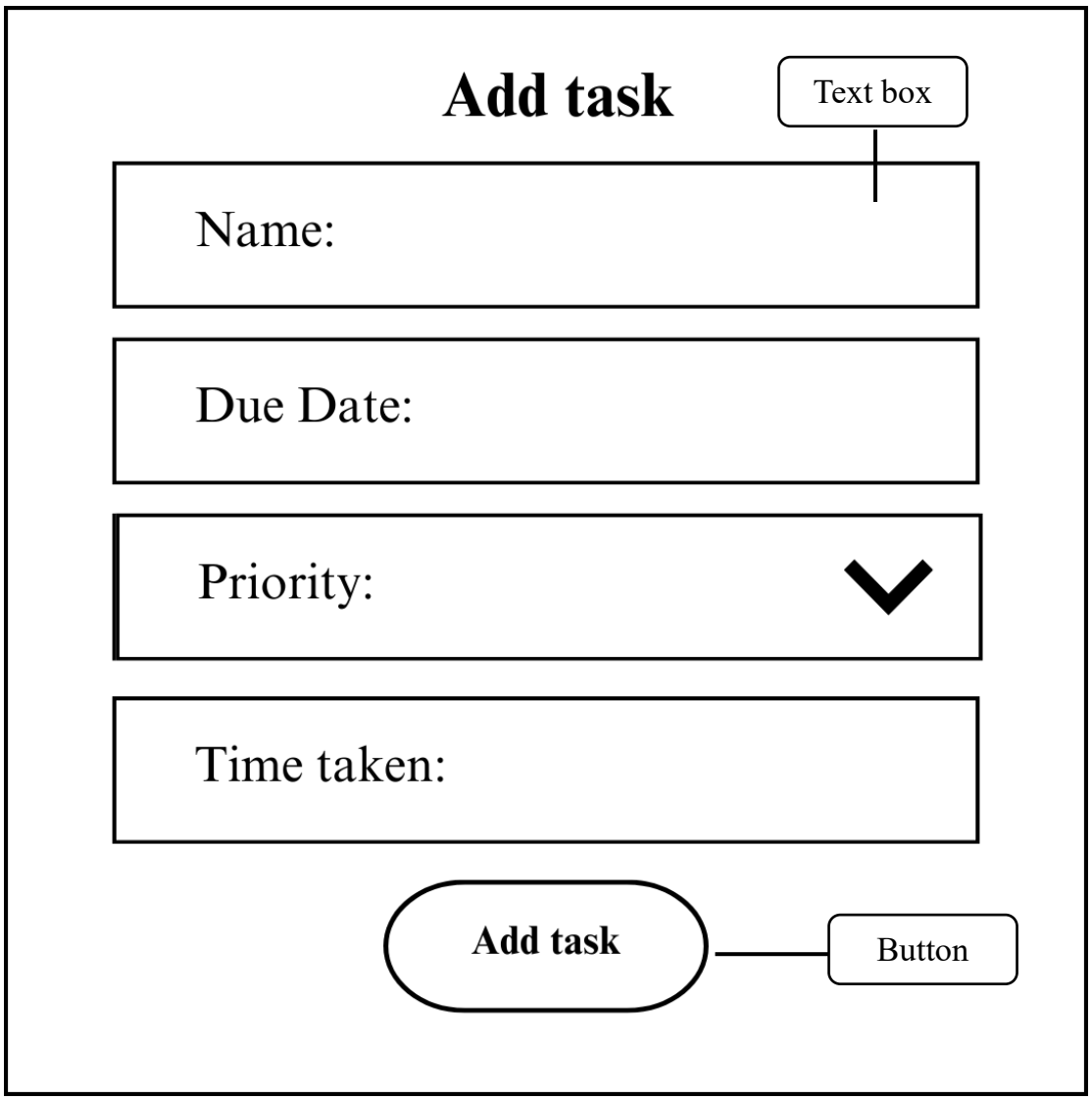

Admin Sign In
Text box
Enter email Id
Enter password
Confirm Password
Name
Key
Submit
Button


Student Log in Page
Text box
Enter email Id
Enter password
Submit
Button
Sign Up


Student Sign In
Text box
Enter email Id
Enter password
Confirm Password
Name
Submit
Button

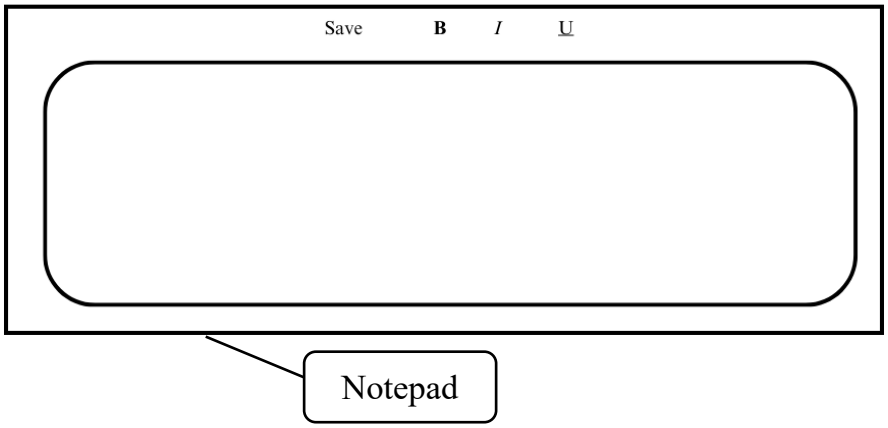

Sign Out
Home Page
Welcome user
Task manager
Calendar
Buttons


Sign Out
Task Manager
Name
Duedate
Priority
Time
Task 1
12th April 2023
High
03:00:00
Delete
Edit
Task 2
12th April 2023
Low
04:30:00
Delete
Edit
Task 3
16th April 2023
Medium
02:00:00
Delete
Edit
Table with data from the database
Add task
+
Buttons
Button

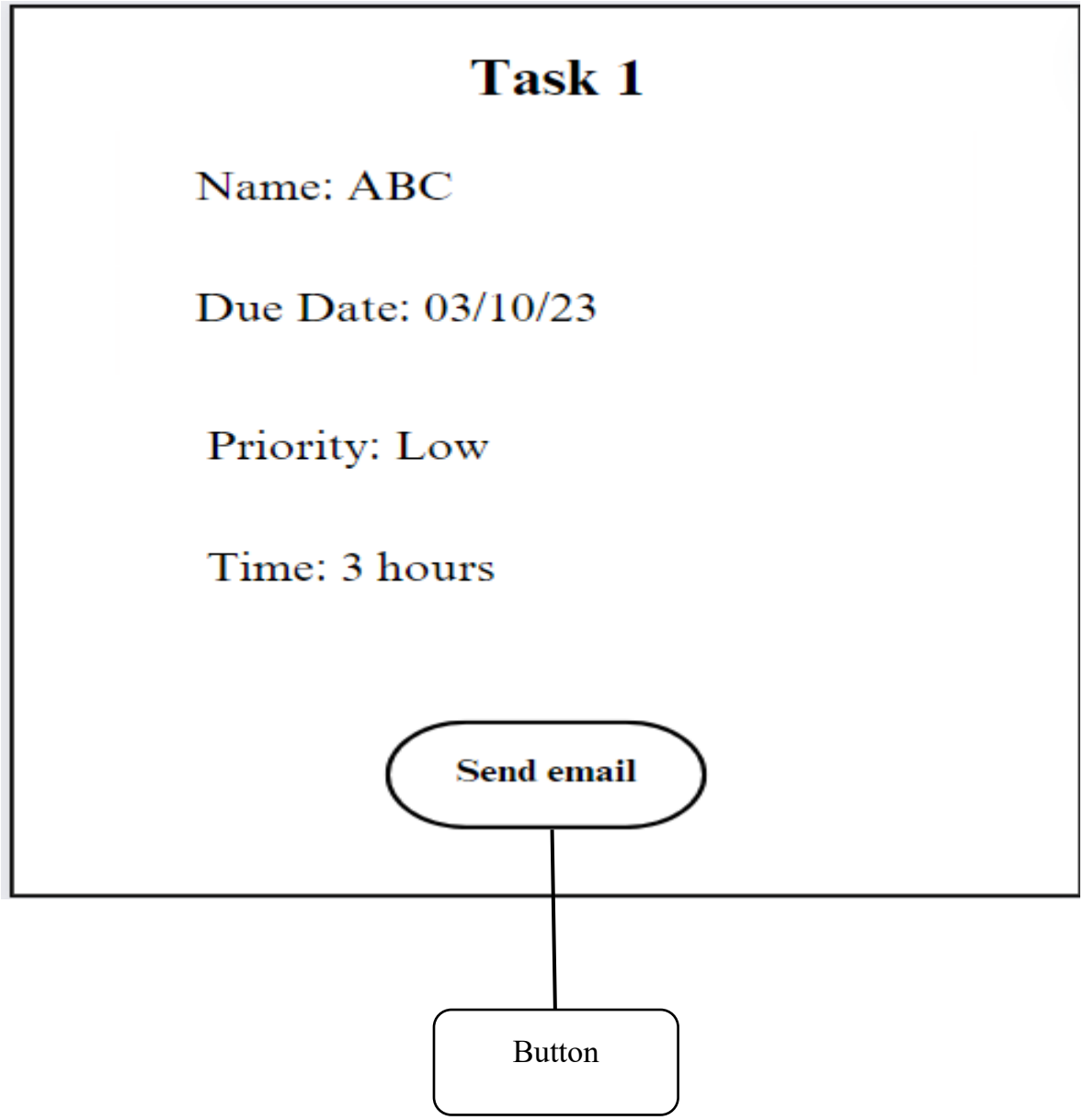

Add task
Text box
Name:
Due Date:
Priority:
Time taken:
Add task
Button

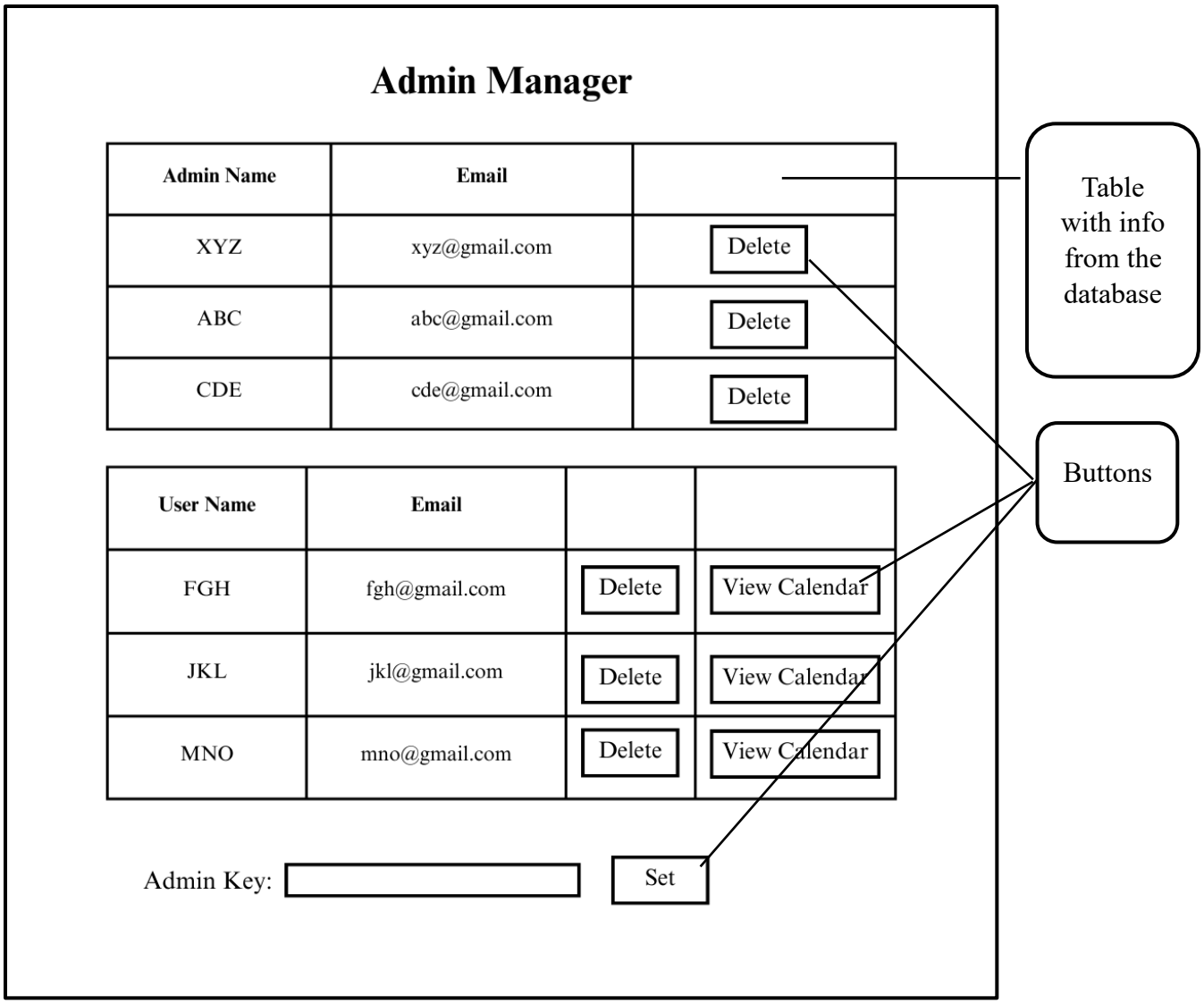
Calendar

				1	2	3
Task 1 4	5	6	7	8	9	10
Task 2 11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29		



Window that appears when an event on the calendar is clicked:




Admin Manager
Admin Name
Email
XYZ
xyz@gmail.com
Delete
ABC
abc@gmail.com
Delete
CDE
cde@gmail.com
Delete
User Name
Email
FGH
fgh@gmail.com
Delete
View Calendar
JKL
jkl@gmail.com
Delete
View Calendar
MNO
mno@gmail.com
Delete
View Calendar
Admin Key:
Set
Table with info from the database
Buttons

Test Plan

Success Criteria	Test Plan	Input	Output
1	A student or admin sign-up is prevented when sign-up conventions are not followed.	Enter an email without the '@' symbol.	An error occurs.
		Enter a weak password	An error occurs.
	A student can sign up successfully	Enter a correct email ID and strong password and press the signup button.	The student is informed about their successful sign-up and their information is entered into the database.
	An admin can sign up successfully	Enter the right details but the wrong key	Inform admin that key is wrong
		Enter the right key	The admin is informed about their successful sign-up and their information is entered into the database.
2	A student or admin can successfully log in.	Enter improper email and password details.	The student/admin is informed about the type of error that has occurred.
		Enter correct details into the fields	The user is directed to the next window.
3	A student can add a new task using the add button	Enter all the details of the task and click submit in the add window form.	The add window closes and the task appears on the window and is added to the database.

	A student can delete a task	Press the delete button for a specific task	The task is removed from the display and the database.
	A student can edit a task	Press the edit button for a specific task	An edit window will appear populated with the current data about that task for the user to edit. Once changes are submitted, they will appear on the app and in the database.
	The tasks are arranged in order of due date and priority	Enter tasks with succeeding due dates and the same due dates with different priorities	The tasks will be ordered by the earliest due date and highest priority for the same due date.
4	The tasks are displayed on a dynamic Calendar	Enter or delete tasks	The Calendar should dynamically change every time a task if added or deleted
5	A functioning notepad	Enter up to 7999 characters into the notepad and press save.	The notepad displays the text and enters the data into the database.
6	Reminders can be sent to student's emails	Click the send email button for a specific task	An email will be sent to the student's address with the correct event details and a reminder message
7	A functioning admin window	Press the delete button for any individual user	That user is deleted from the display, as well as the database
		Enter a different key in the admin window, log out, and sign up as a new admin.	The admin should only be able to sign up using the updated key.
8	Admin should have access to student Calendars	Press the View Calendar button for a specific student	The Calendar of that specific student appears.

Proposed Algorithms

Libraries that I plan to use:

FullCalendar – for the Calendar as it works the best with node.js environment and on different Operating Systems. It also allows for customization of events on the Calendar.

Quill Editor – for the notepad as it is the most compatible with Electron, and is known to provide built-in CSS files allowing for easy customization with a base already set.

Node emailer – for sending emails as it facilitates an SMTP connection well with the least amount of requirements while still ensuring a safe connection from apps to email.

MySQL – will be used to connect the electron app to the database, as it is one of the easiest and fastest ways to do so, even for students who would need to slightly modify the code to match the MySQL database installed locally on their systems.

Electron APIs:

For front and backend interactions – Makes verification and validation of forms during login easier.

Use the built-in notification class to send notifications. This is much easier than a library and because it's electron provided, it can adapt depending on the OS of the user's laptop.

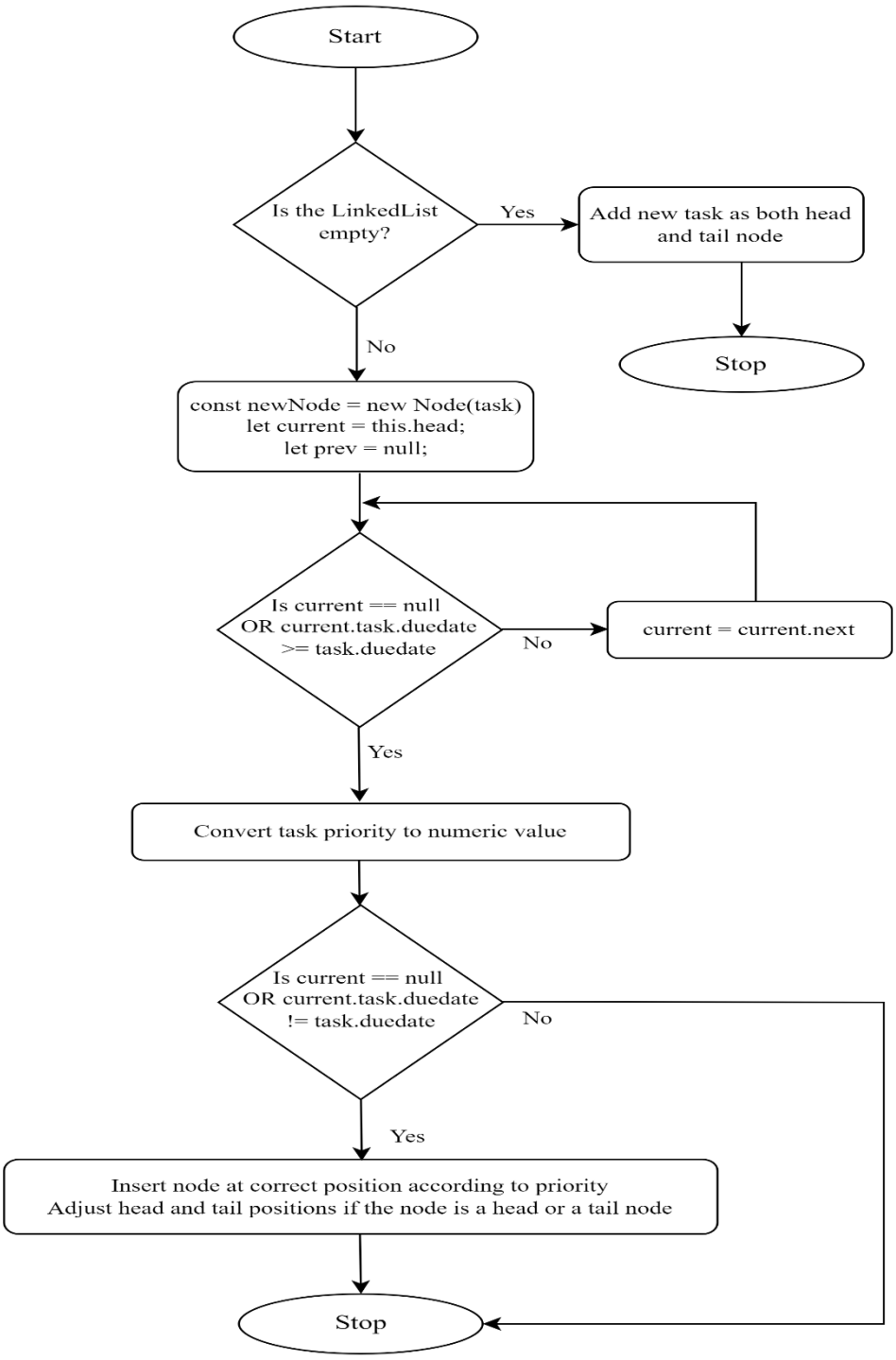
CSS:

The client chose shades of purple, teal, and blue as according to her research they induced the most productivity, hence most elements made from scratch will be set to those colors using CSS.

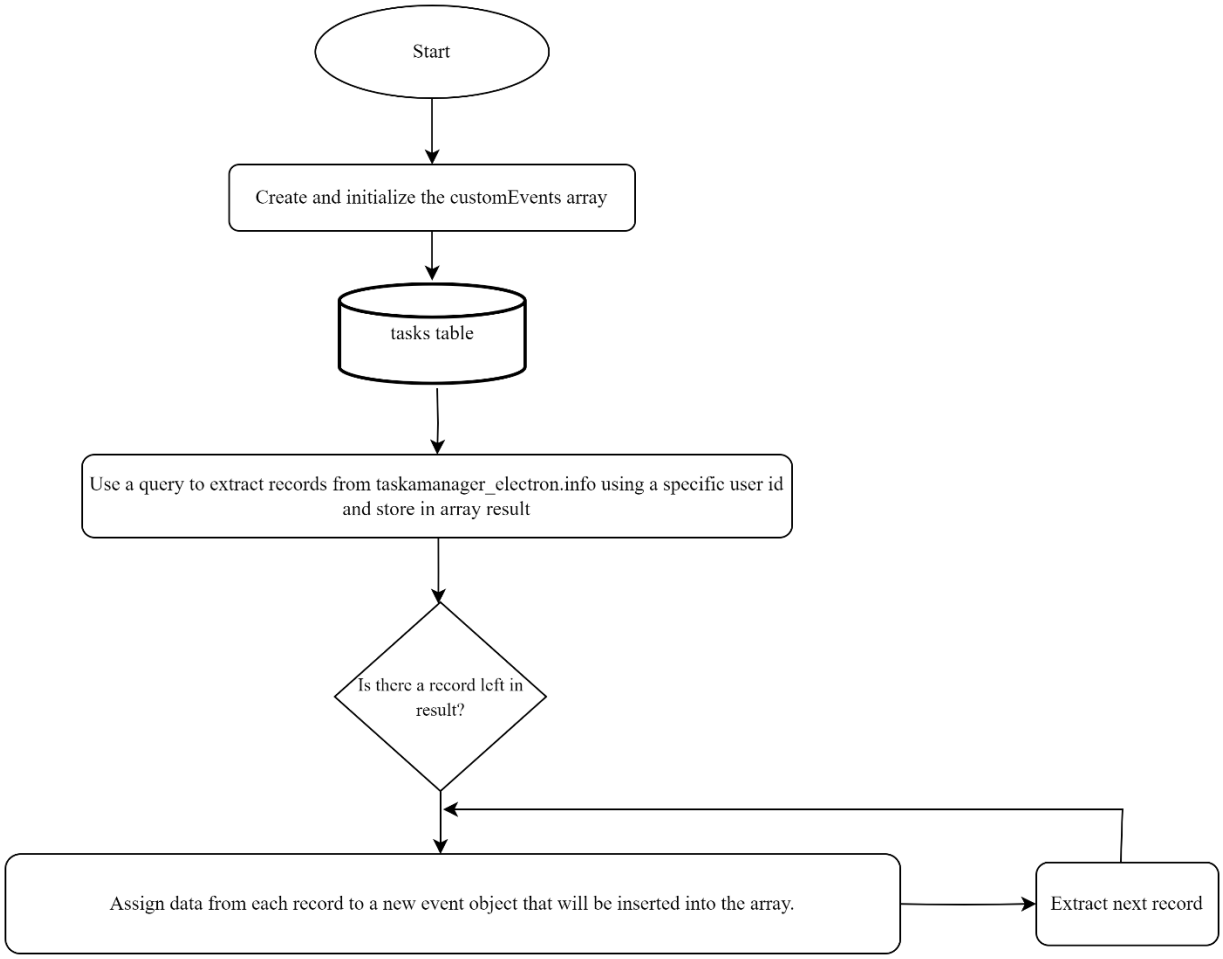
However, the libraries I will be using will require imported CSS style sheets – especially the Calendar and the Notepad. I will be altering the color schemes of the style sheets while retaining the shape and form of the elements.

Flowcharts

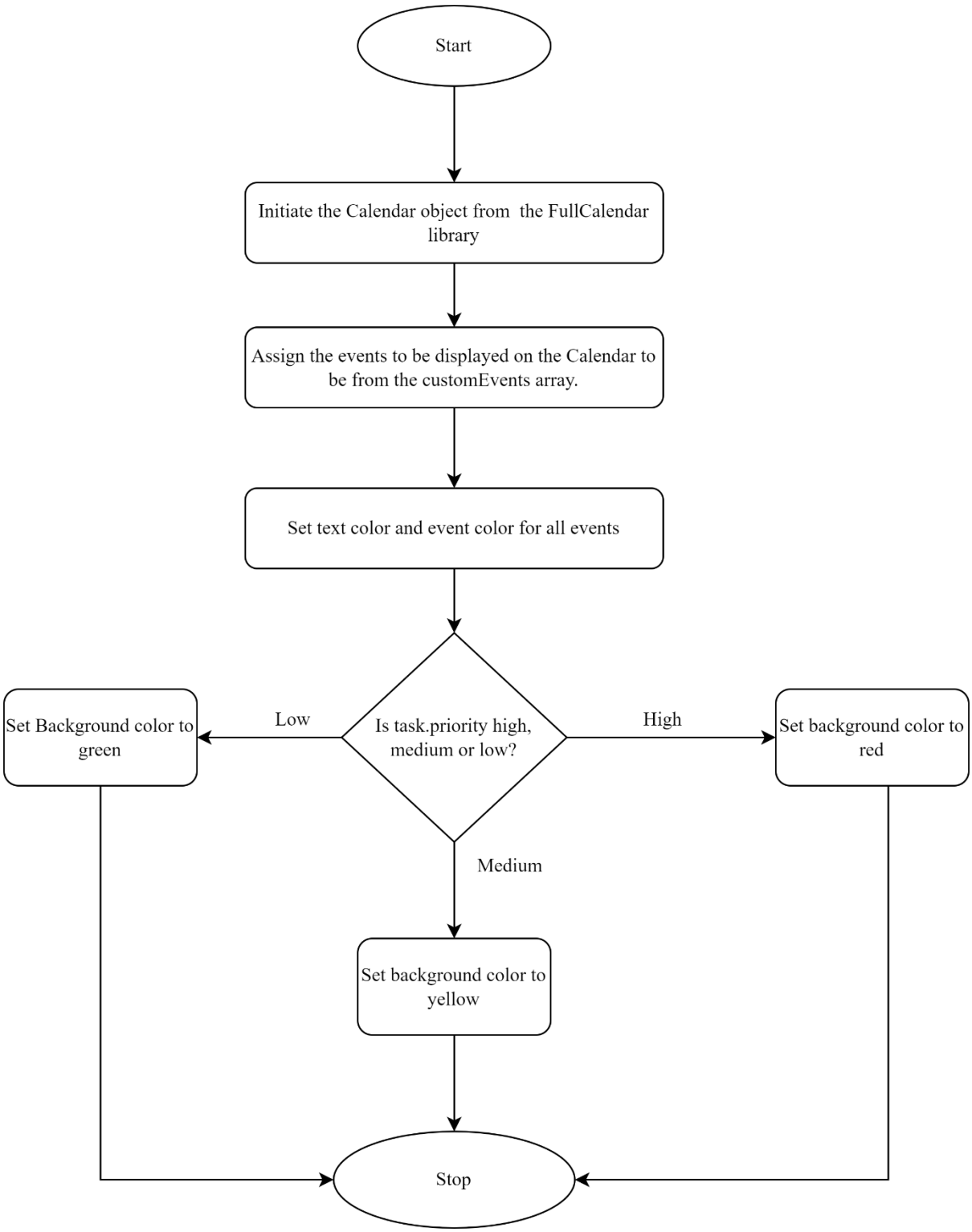
1)Arrange tasks in a schedule format using the highest priority and closest due date.



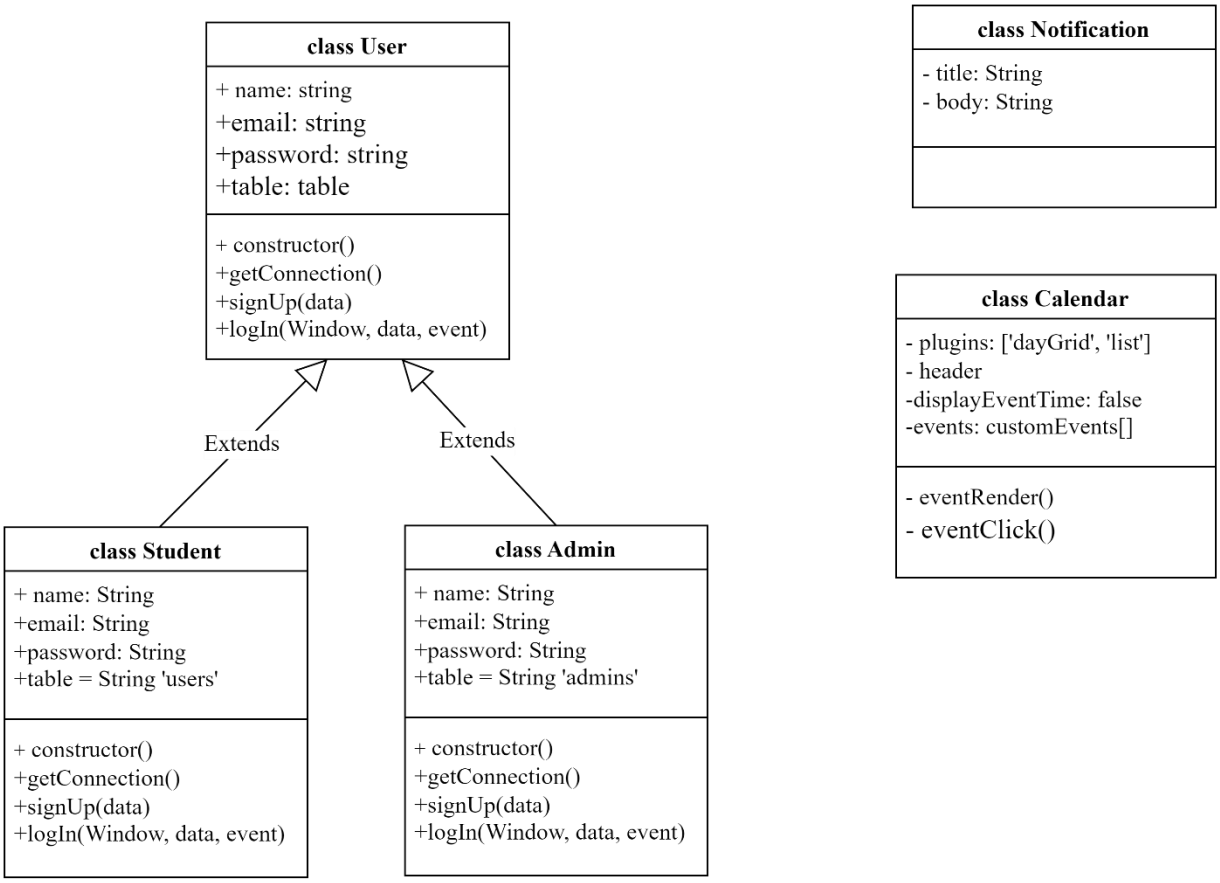
2) Creating events for the Calendar by extracting data from the dataset.



3) Displaying the events on the calendar according to priority



UML Diagrams



Data Structures

Users/ Admin Table

Field Name	Data type	Size
User/adminid	INT	100
email	VARCHAR	100
password	VARCHAR	225
name	VARCHAR	100
key	VARCHAR	45

Notes Table

Field Name	Data type	Size
userid	INT	100
notes	VARCHAR	8000

Tasks Table

Field Name	Data type	Size
id	INT	100
name	VARCHAR	200
due date	DATE	
priority	VARCHAR	200
time	TIME	
userid	INT	100