Web Based Utility Application:

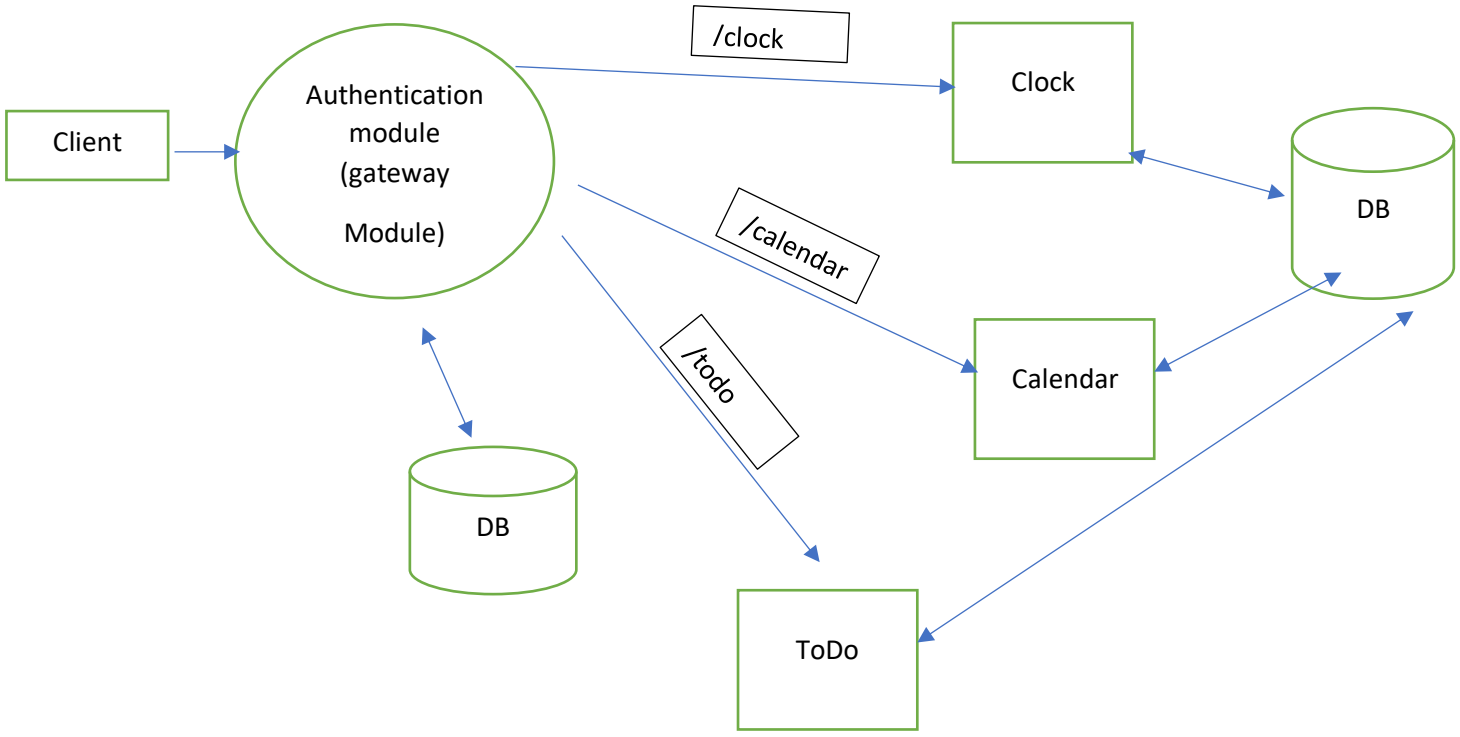
Overview:

The main idea of the proposal is to bring the basic utility apps together under one hood. We have faced many difficulties in the basic utility features[clock, calendar, Task]. The project addresses certain hinderances in the basic app functionalities like sharing alarm, views of tasks in multiple time zones, easy switching between time zone for a better view of things. We are also establishing a good connection between the ToDo, calendar and Clock.

The service implementation in this project is based on the concept of microservices framework. We will have authentication and contacts as a common module. The contact will carry only the email address. The authentication mechanism is needed to keep track of user data across the apps. So, sign-up module is mandatory for us. We will have WUID[Unique ID] for the users. The clock, calendar and ToDo is considered as a separate module which will have the provision to get connected through external endpoints and internal API's.

The contact module will be the heart and it can be reused for all other apps. So basic features for contact modules are given. Users can add or delete contact, and this is persisted in the relational database.

A rough sketch of the framework can be given as below.



Tools to develop the project:

Web Framework: Spring boot
Database: MySql, JDBC adapter
Programming Language: JAVA
IDE: Intellij
VCS: Git
UI: HTML, CSS, JavaScript, Thymeleaf
Build: gradle
Task management tool: JIRA
Testing tool: Junit
Cloud platform: Heroku

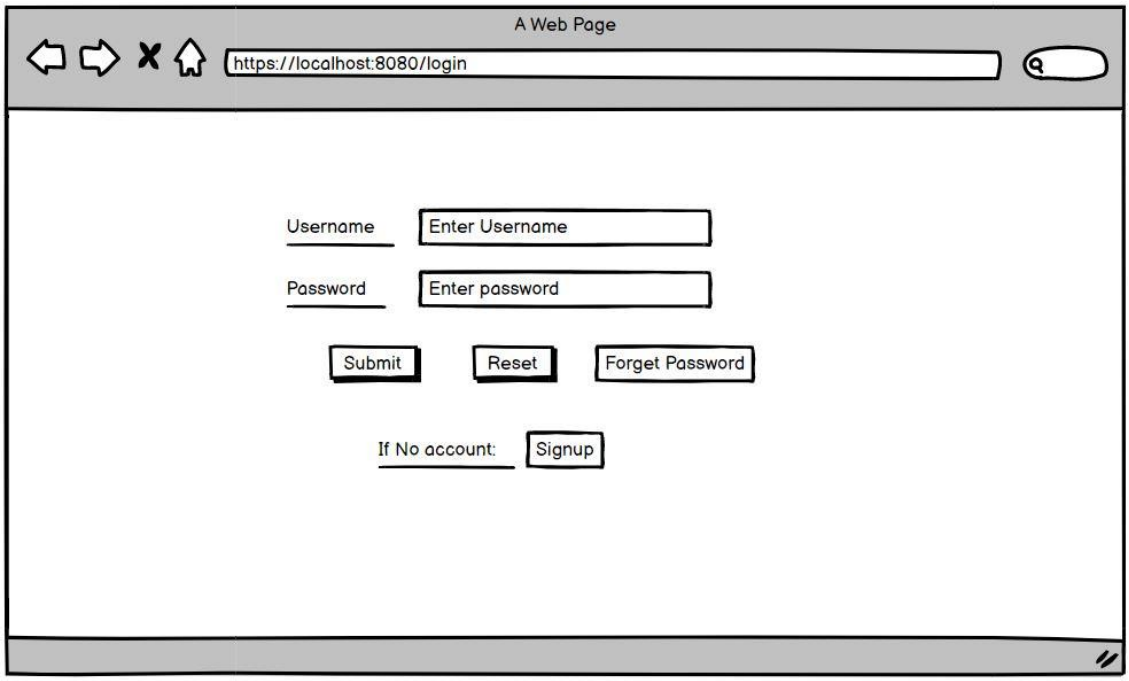
Authentication Module:

The authentication module consists of login page by default and a hyper link to the Signup page.

Login page:

The Login page consists of username[email Id] and password with minimum of 8 characters.

If the user entered wrong email address or invalid email an error will be thrown as “Invalid Username”. If the password has less number of characters or wrong password an error will be thrown as “Invalid Password”.



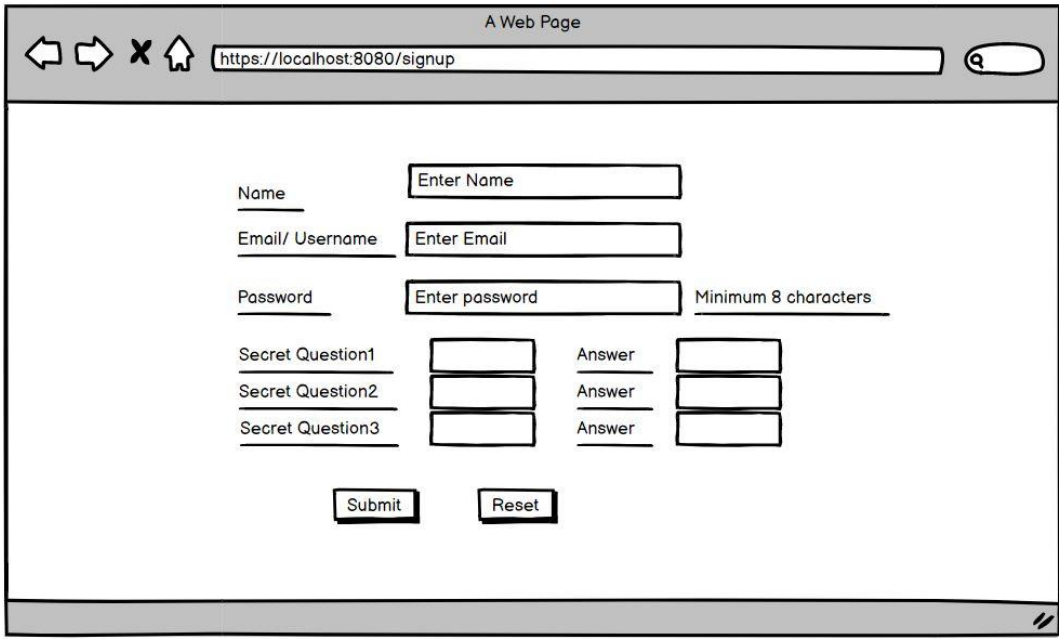
User Login Page

Sign Up Page:

The Sign Up consists of Name, Username[email] and password with minimum 8 characters.

If any of the criteria is not matched an error will be thrown.

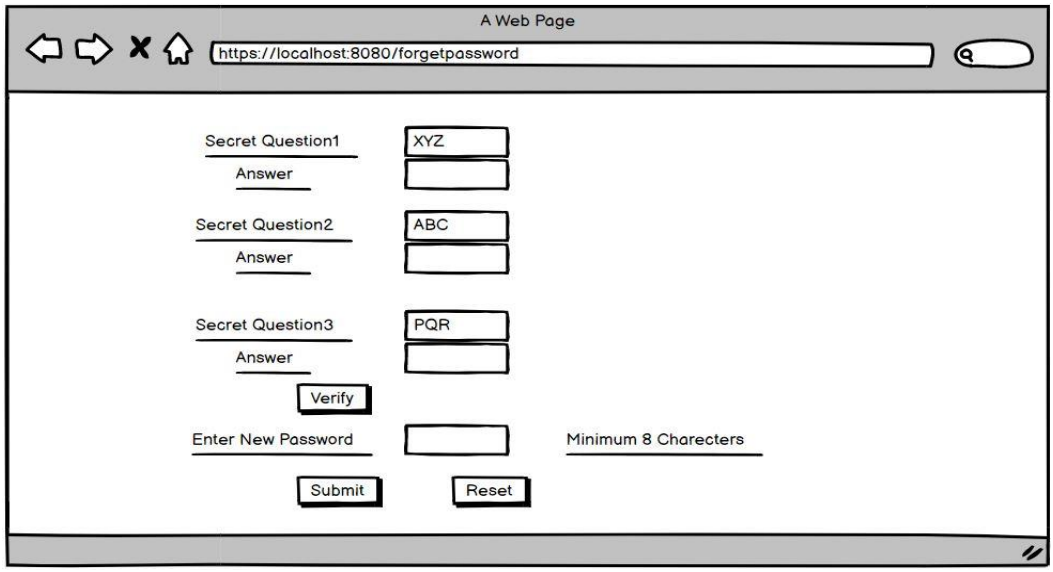
If the standard email criteria is not met, an error will be reported to user as “Invalid Email”.
If the email already exists an alert will be displayed to user as “Email/username Exists”.



User Sign Up Page

User Forget Password:

When the user clicks on forget password, he must answer the security question that he configured during the sign-up process. The answer is verified and then user is allowed to create a new password.



User Forget password

ToDo

We are going to provide functionalities in ToDo as listed below.

Create a Task. [Title, Description, due date, Supporting Attachment]

- User can create a task with the details like Task Title, Task's full description, its due date and attachment of the document.
- User can set the due date according to the time zone.

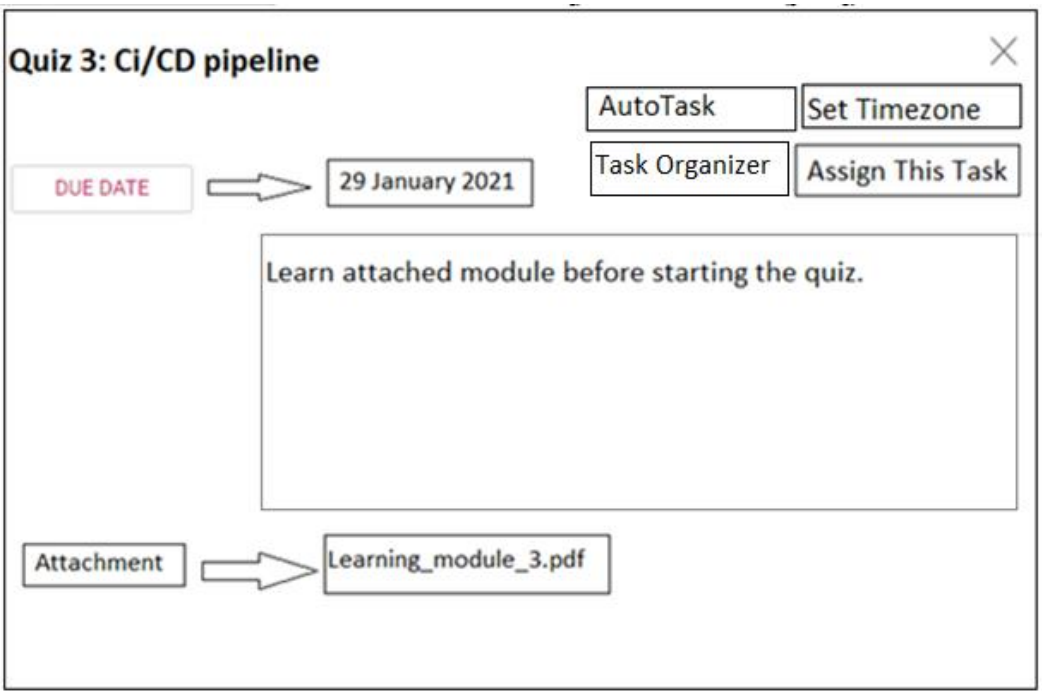


Fig.1

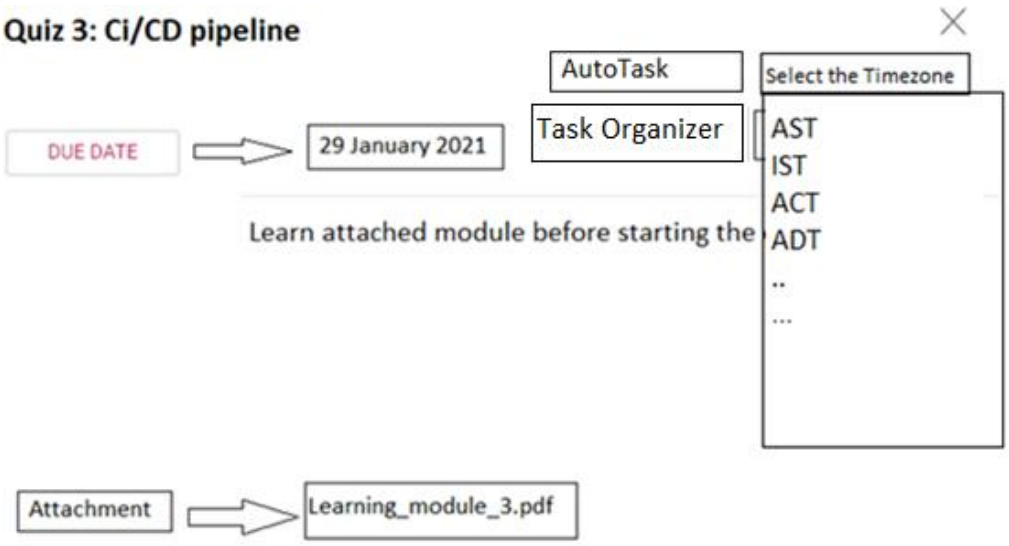


Fig 1.1

- When user press the button for creating the task, Fig 1.1 as demonstrated above will open and after fetching all the details from user it will store it in database.
- After selecting the Time zone, system will automatically convert the time into user's local time and set the due date accordingly.
- If user set the due date older than current date then user will get warning.
- User is only allowed to attach files that are less than 2 Mb in size. If it is not then appropriate error will be thrown.

Create lists of Task.

- User can create a list of tasks for a particular course/topic.

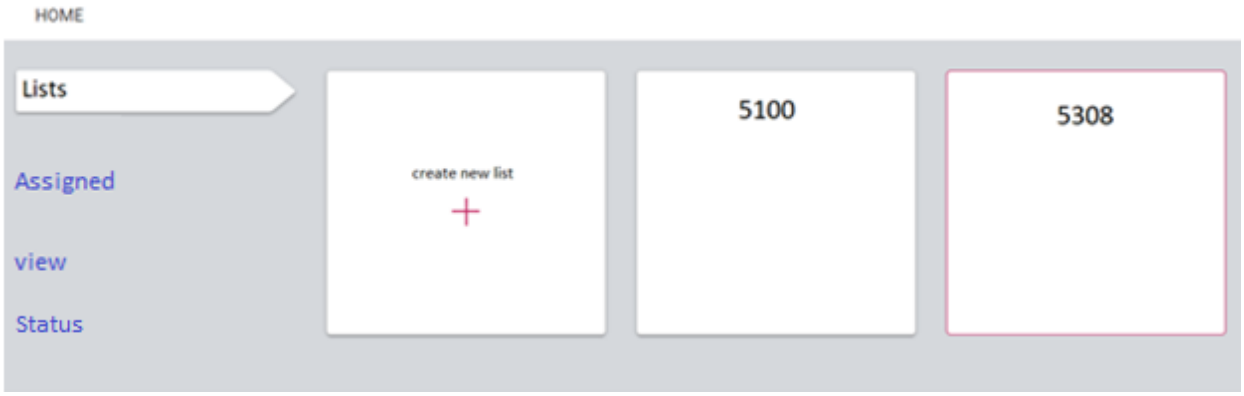


Fig.2

- After clicking on list user can see the tasks associated with that.

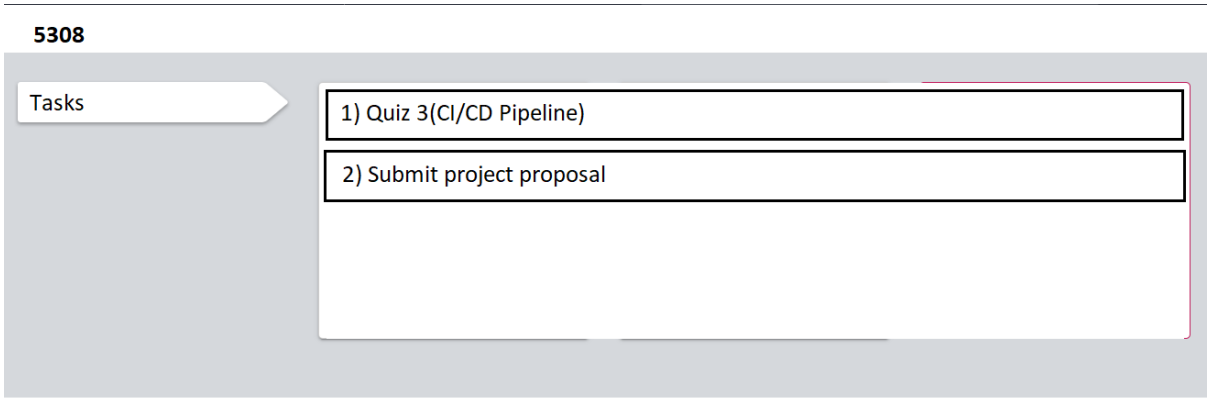


Fig.3

- Data structure like list will be created and it will contain all the task belong to that list. After clicking on a particular list user will be redirected to the screen like Fig.3 and data associated with the list will be displayed to user. After clicking on task in the list, User will be redirected to the screen like Fig.1 and data associated with the task will be displayed to the user.
- When user will Create the lists with the same name then he/she will get an error message.

Assign user to the Task.

- User can assign the task to another user by simply selecting other user's contact information like email Id.

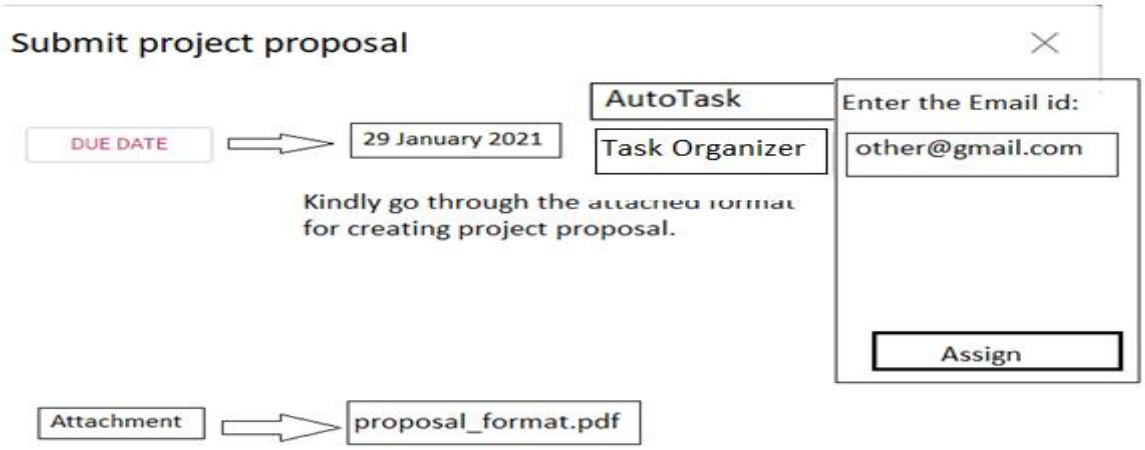


Fig.4

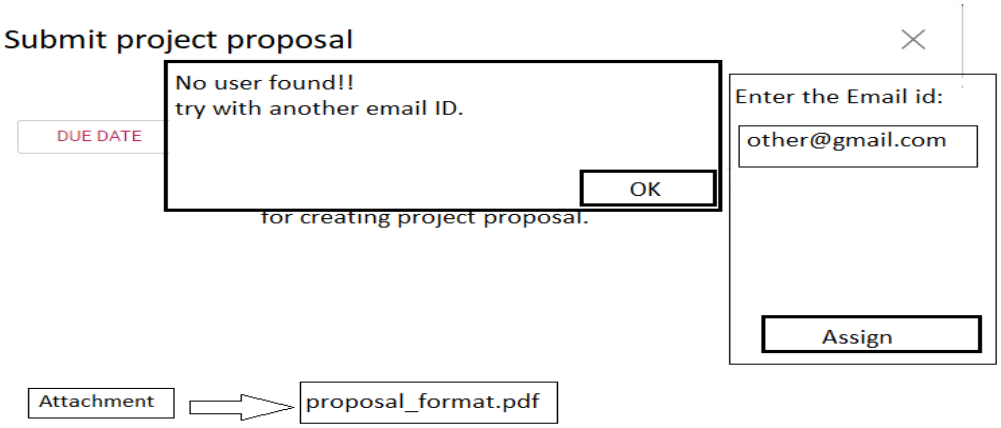


Fig.5

- After pressing the Assign button as per Fig.4, System will search the other user with provided email ID and if that email id is registered in database then the task will be reflected in the other person's task list and system will set an authority to other person for modification of the task. System will update the other person's account like setting the due date for assigned task in his/her account. System will store the details of Assignee, Assignor and Task
- If email id provide by user is not registered in database, then system will alert the user.

Options to Modify/Delete the Task and set the status of the task

- User can modify the details of task provided after task creation and can delete the task.

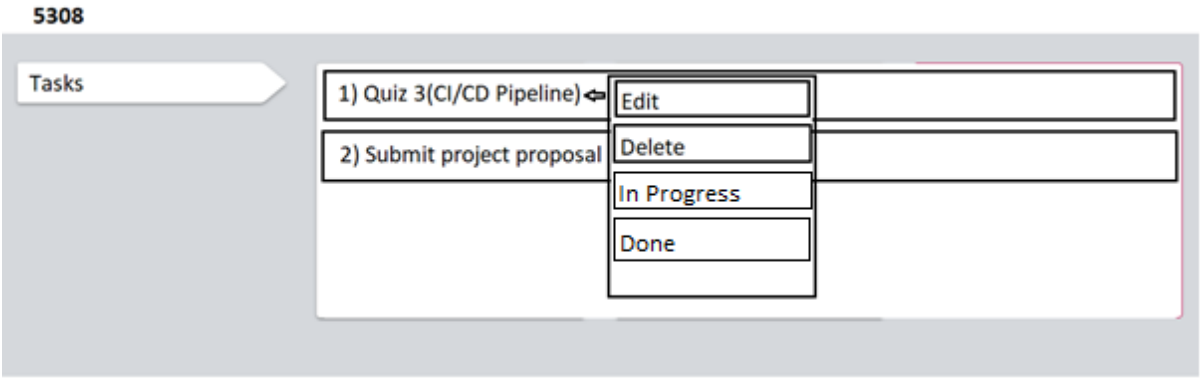


Fig.6

- When an edit is made, the data is appropriately modified in the relational table by communicating with the persistence layer from the application
- System will delete the entry of particular task and if this task is assigned to other person then system will also delete the same task from other person's account and vice versa. these changes will be reflected in the status.
- System will remove the task from the list and put it in status based on the user selected option like in progress state or in done state.

Views in ToDo.

- User can view the all tasks as per applying filters like sorting with due dates, just view the assigned tasks, view the all task together.

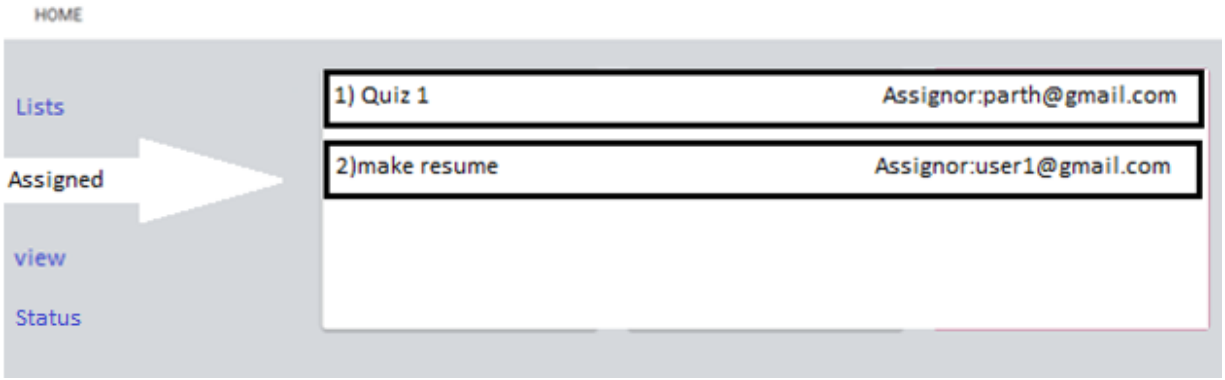


Fig.7

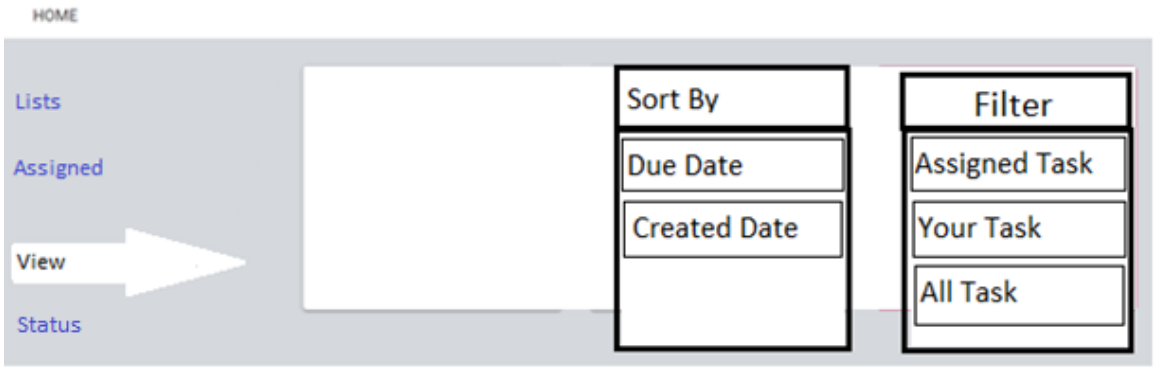


Fig.8

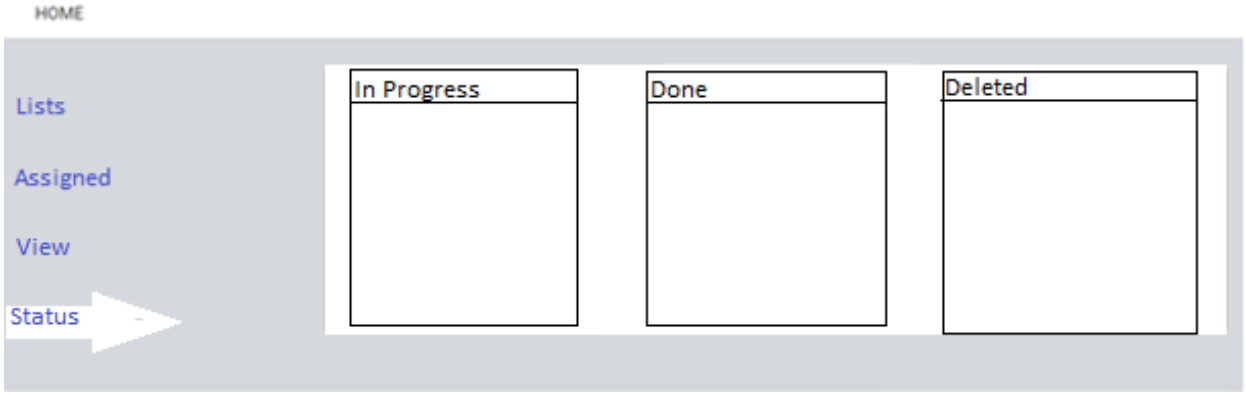


Fig.9

- After clicking on Assigned system will display the result of assigned task along with the assignor by fetching details from the database.
- After clicking on view button system will fetch the result based on applied filter and sorting.
- If there is no task recorded in user's account then system will show message like 'No task found'.
- System will move the task from the task list to the status and put the task in one of the categories based on user action like delete, In progress or Done.
- If assignor change the status of the task then the changes will also be reflected in the assignee's account and vice versa.
- If assignee and assignor try to change the status of the task at the same time then error message will be displayed by system.

Detecting the task recurrence based on key words.

- User can set the task by just providing keywords.
- The recurrence of Task will set automatically after fetching the task description.

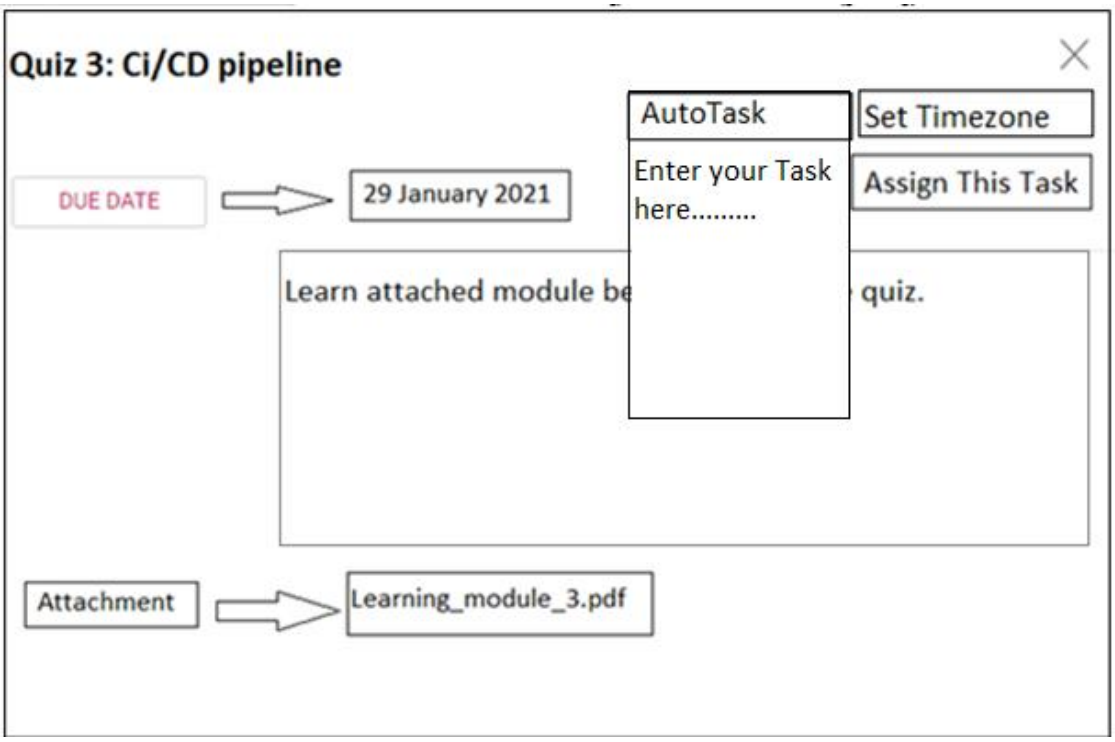


Fig.10

- System will take the input string from user and based on the keyword found in the string, set the automatic task for user.
- System will use string matching algorithm with the words in the dictionary.
- System will show error message after entering invalid string by user.
- If system will not find any meaning of text entered by User then it will simply alert the user by message “cannot find context of the keyword. please try with different keyword”

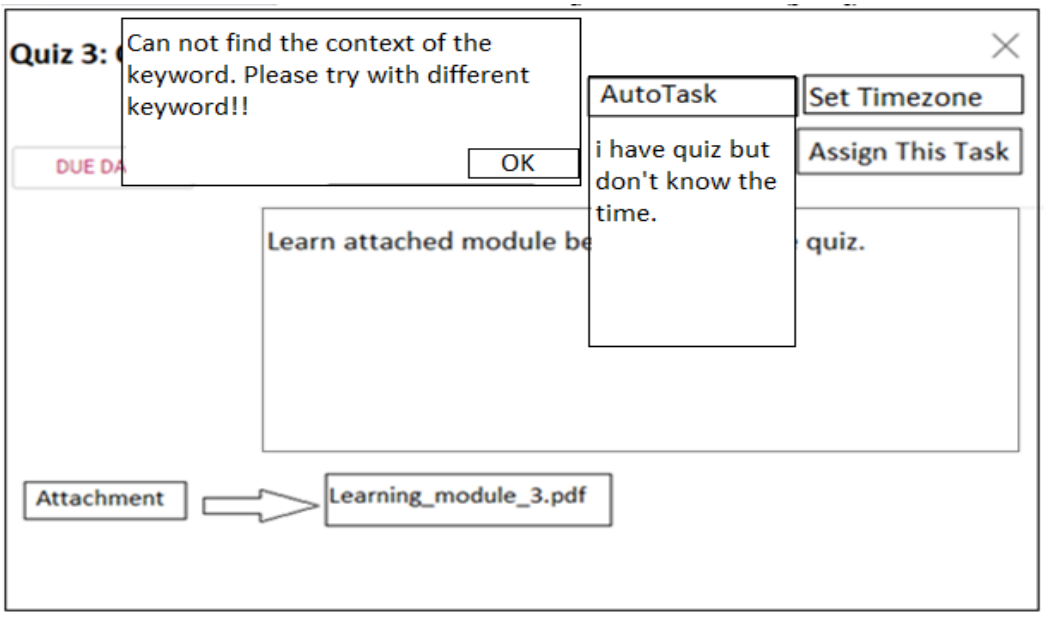
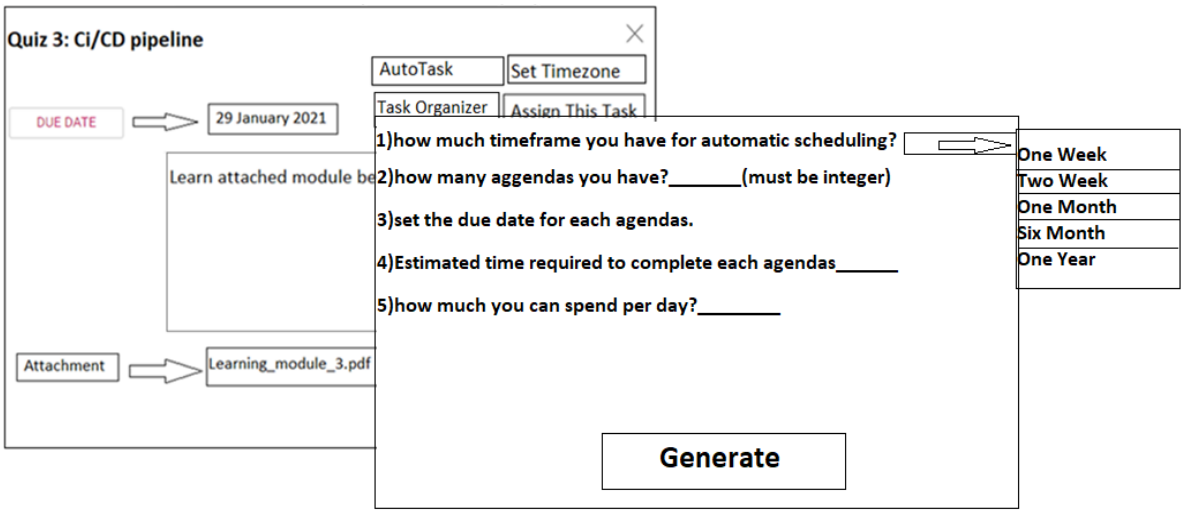


Fig.11

Automatic Task Organizer [ATO]

- User can organize the task after answering few questions.



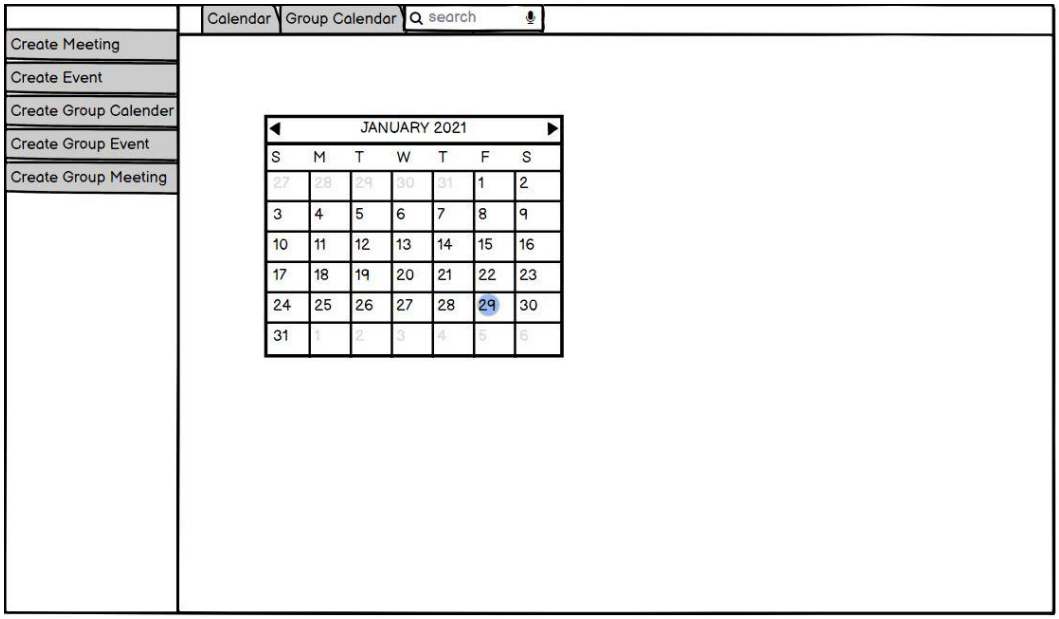
- System will organize the task schedule as per user's answer by performing priority scheduling algorithm.
- system will show an error if user will press generate button without entering anything in the form.
- System will show error message after receiving incorrect or illogical user input.

Calendar

Calendar View:

When the user select calendar from the landing page, the default individual user calendar for the current month will be displayed, the user can have options to check other month and year.

There will be a Menu bar at the top where the user can navigate to group calendars. When the user clicks a day in calendar it will redirect him to see the list of events and task scheduled for the day for both group calendar and individual calendar.



This is a Default Calendar View

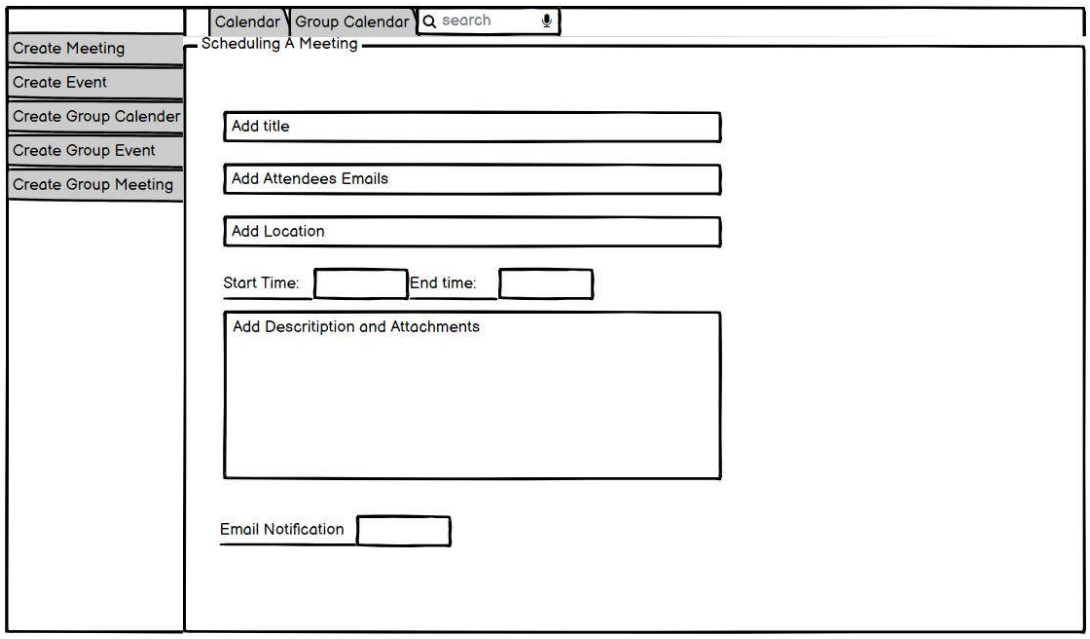
Scheduling a meeting:

User can schedule a meeting with other users. The user can select the Create Meeting from the side menu bar.

A form will be displayed and the user will have multiple fields like Meeting Agenda, Attendees Email, Location, duration and meeting notes with attachments.

Once the form is submitted the data will be passed to the application layer and stored in the database mapped to the users using emails, after storing the data a meeting invitation will be sent to attendee's emails and also the meeting will be added to the user calendar. Additionally, an email notification time can also be set, default will be 30 Minutes before meeting, but the user can modify it accordingly.

If the meeting is not successfully created then the user will get an error message in screen.



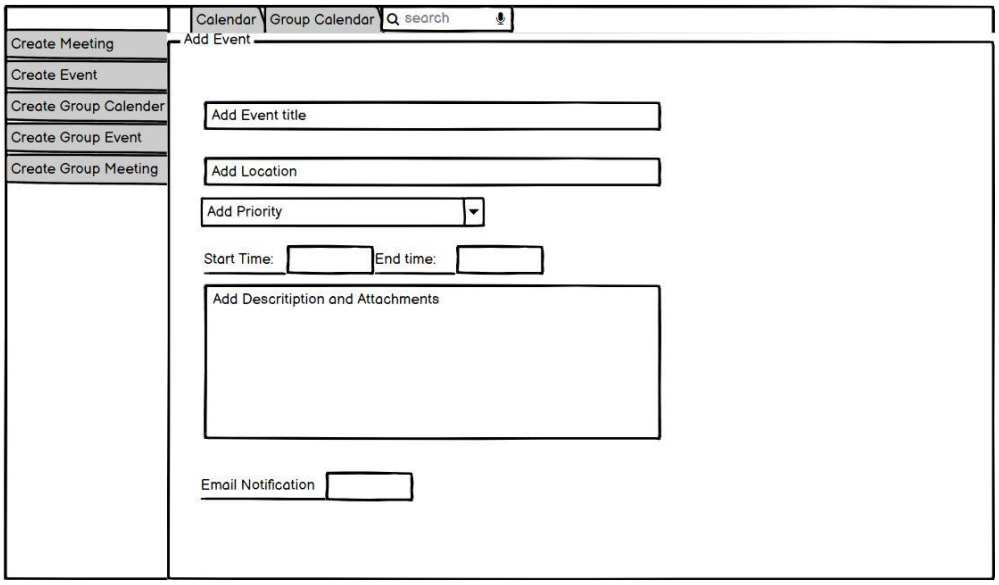
User Scheduling a meeting

Adding an event to calendar

In this feature User can create an event to the Calendar based on the input. The event will provide the user with multiple options like Event Title, location, duration, Priority and when to give an email notification.

On the submission, the data will be processed and stored in a relational database for the user and once successfully stored it will update the calendar with the respective event.

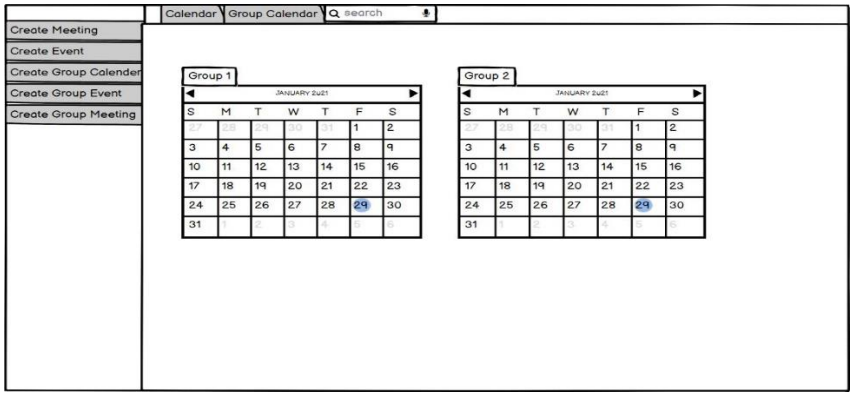
If the event is not successfully registered, then the user will be alerted as “The create event failed”.



User Adding an Event

Group Calendar

The user can select the Group Calendar from the menu bar. The calendar has the features that can create a Group Calendar and Create group event or group meeting.

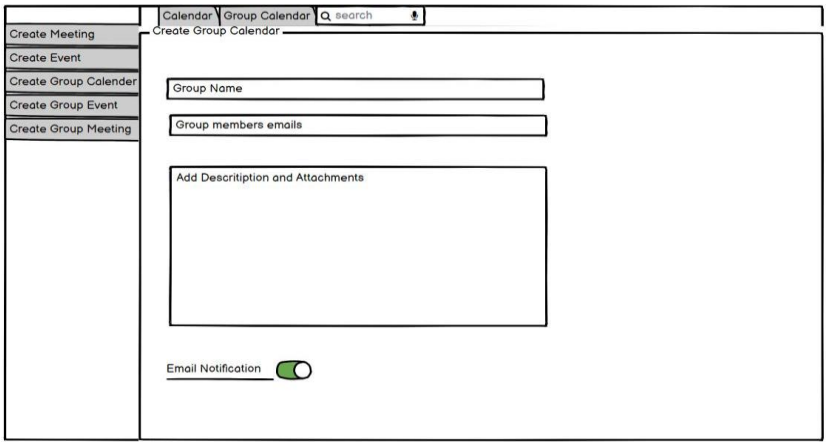


This is the Group Calendar View

1. **Creating a Group**

   If the user selects to create a group calendar, then the user will be redirected to a create page where a form will be displayed. Once the user submits then a new table will be added to the database with the listed members and a new group calendar will be added.

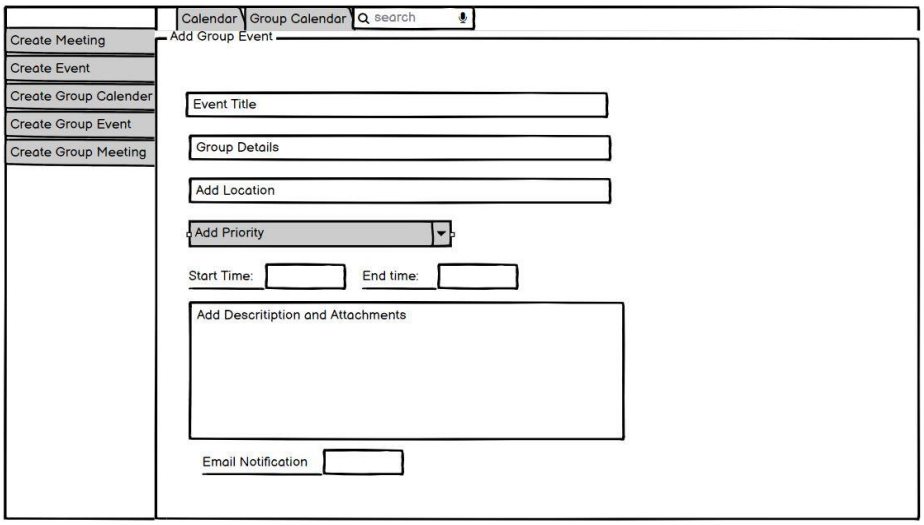
   If the transaction is not complete, then a message creation failed will be displayed to the user.



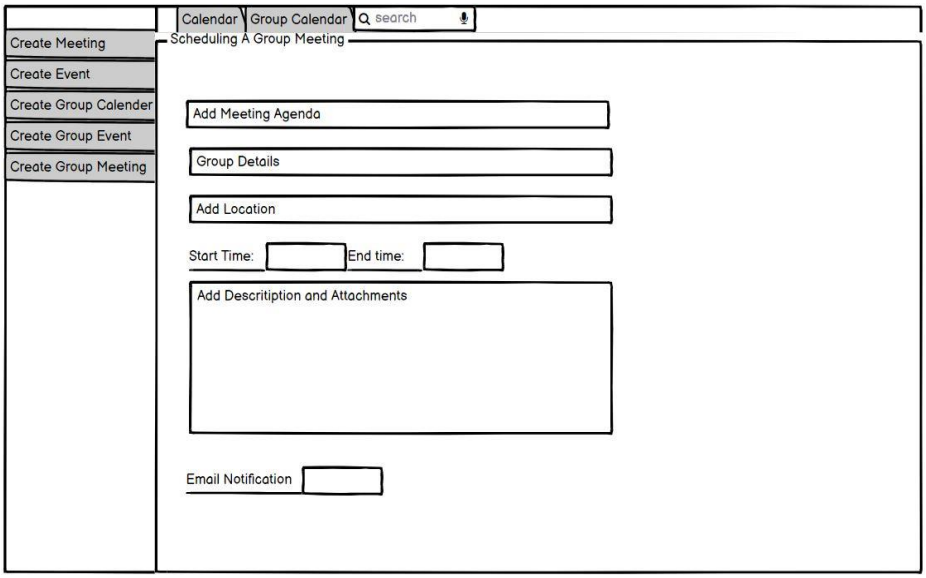
User Creating a Group

2. **Adding Event/ Meeting to group**

   If the user selects for a Create group event or meeting then it will redirect to a form to provide details like Agenda, Group Name, Duration, Add text and user clicks on submit. On submit, the data will be processed and update the group table, on successful transaction an event/meeting will be added to that group calendar.
   If the transaction is failed then message for user will be displayed “Adding of event/meeting failed. Please Try Again”.



User creating a Group Event

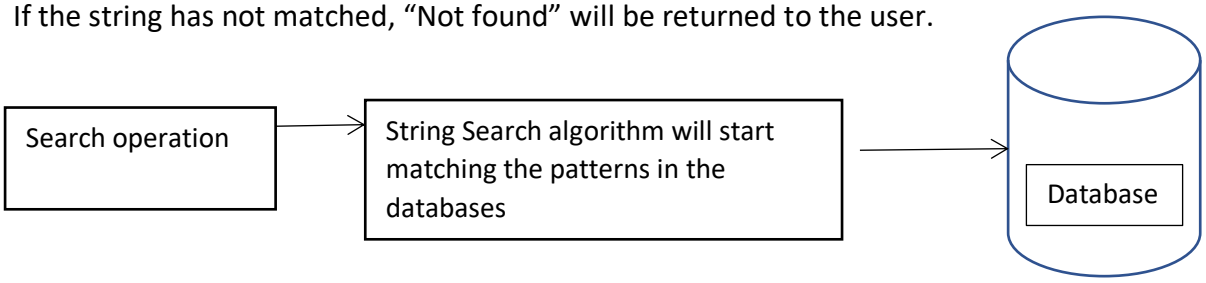


User creating a Group Meeting

Search operation in calendar:

If the user doesn't know all the details of an event or mail or when it is scheduled, then user can use search option to type a keyword. Once the user entered the keyword, then the string is taken as input and a string search algorithm will be applied on the existing events and meeting in database and the results will be presented to the user.

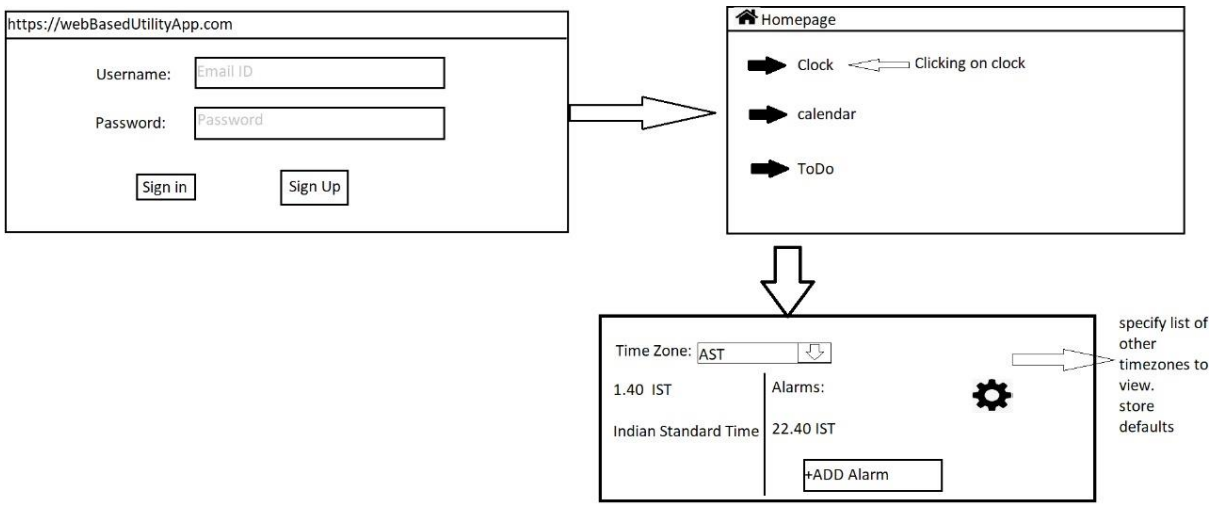
If the string has not matched, "Not found" will be returned to the user.



Clock

Clock will have its own set of tables in the provided database. Below are the functionalities and implementation of each.

The landing page after sign-in will have clock as a list item. When user clicks on Clock button, they will be redirected to a URL https://<website>/clock . The mapping will be established using SpringBoot.



Landing page

View page: [Display clock of the user preferred time zones]

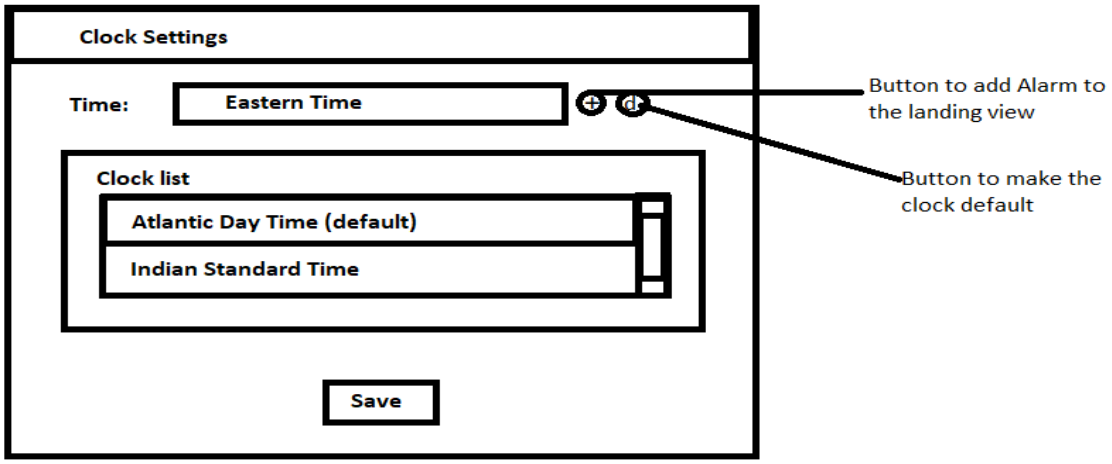
It will have the current time shown. User will be provided to set as many clocks in the view as per time zone.

First time user will be asked details like Default Time zone and will be allowed to add 3 or more time zone for the convenience.

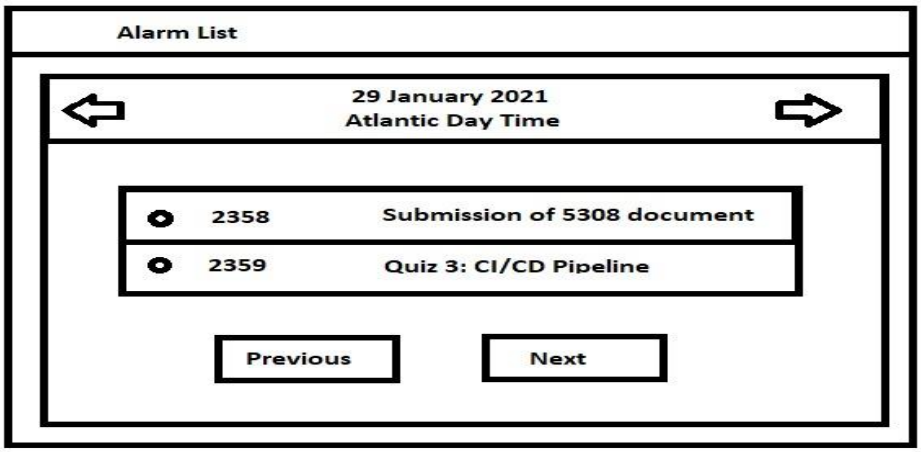
The preferred time zone will be stored in the database for loading the view properly for the next time. The UI will be a form containing those fields and when the form is submitted the data will be passed to the application layer and stored in the database mapped to the user. Before storing the data in the database proper validations are done.

Along with the current time details, list of alarms will also be displayed for the user. Few more detailed information like time left for next alarm is shown in the page. We will use simple ordered list to display the information. Those details are fetched from the server on page load in a json object. Proper views are designed using Thymeleaf and redirected from the server on request.

To display the current time in the view minimal java script will be used.



Clock Settings



Alarm List

Alarm:

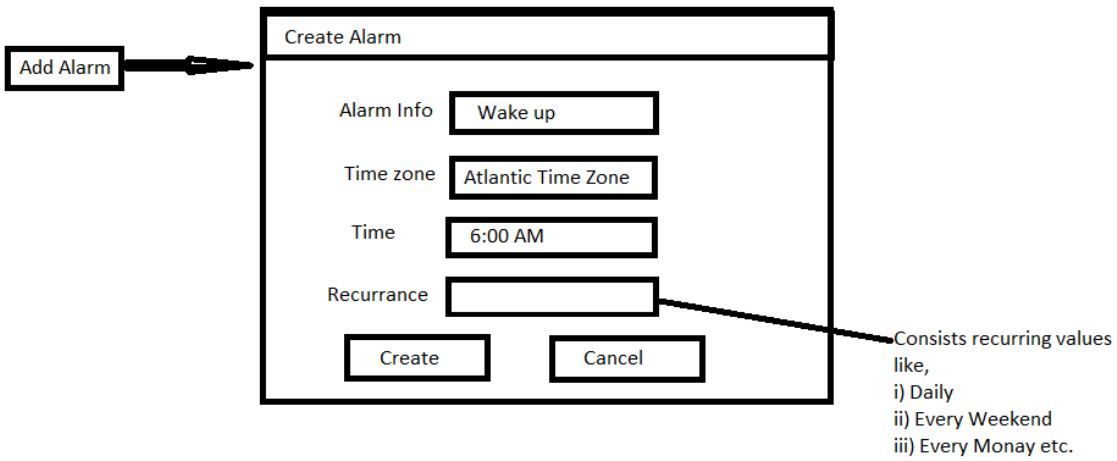
Basic functionality of any Clock app is to store the alarm. User will have a button in the web page to create alarm.

Create alarm button. →

A form is displayed to fetch input, inputs will be:

i) Time.
ii) Contact or individual for whom the alarm has to be set.
iii) Time zone in which the alarm has to be set.
iv) Snooze duration
v) Purpose of alarm
vi) Recurrence (Prediction based on text in alarm purpose)

→ POST: /clock/addAlarm will be called from the UI on Submitting. Then the Alarm object will be persisted in the database under the user.

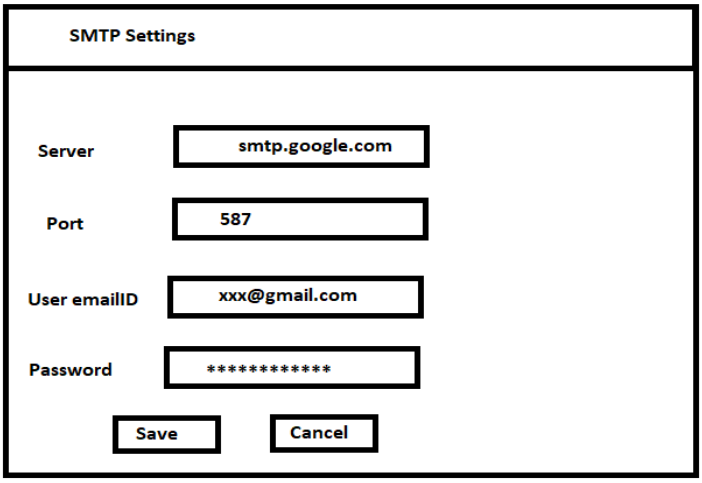


Create Alarm

On clicking Create button, the alarm information is successfully persisted in the database.

We will send an email at the alarm time to notify the user. To send an email we will implement the connectivity of SMTP on the website. This will give access to send emails to the user using Java Send Mail. Before storing the SMTP settings an email is sent and configuration is verified. If there is a failure proper error message is displayed on the screen.

Before adding alarm, the server makes sure the saved alarm time is not already passed.



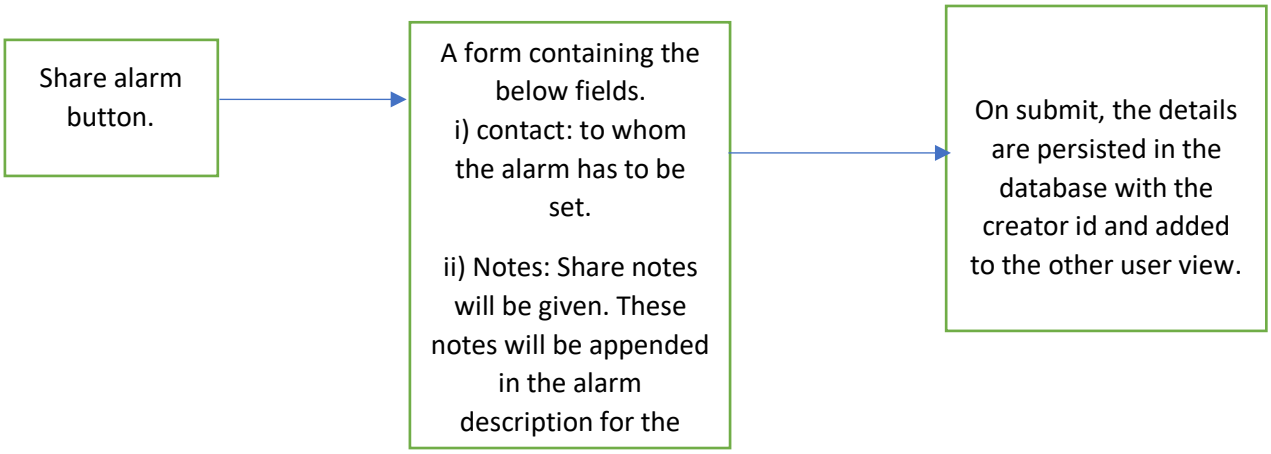
SMTP Settings for sending an email

Share Alarm:

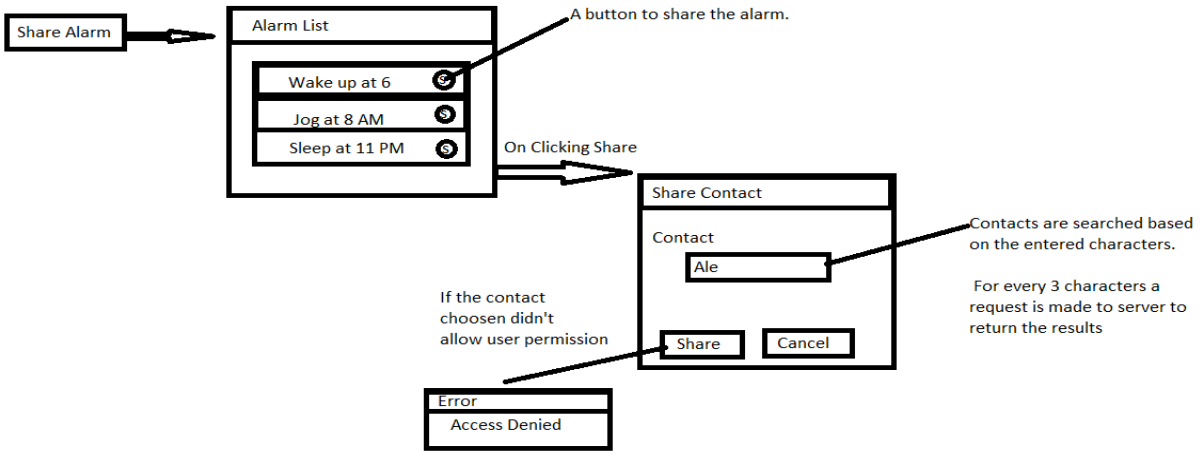
If someone wants to share their alarms to another user an option is provided even after creating the alarm.

In the list of alarms displayed on the home page, when user clicks on the alarm the detailed description of it is provided for the user. When the user clicks on “Share” button then the user contact will be loaded.

When user selects one contact then the alarm will be added to the other user in the database.



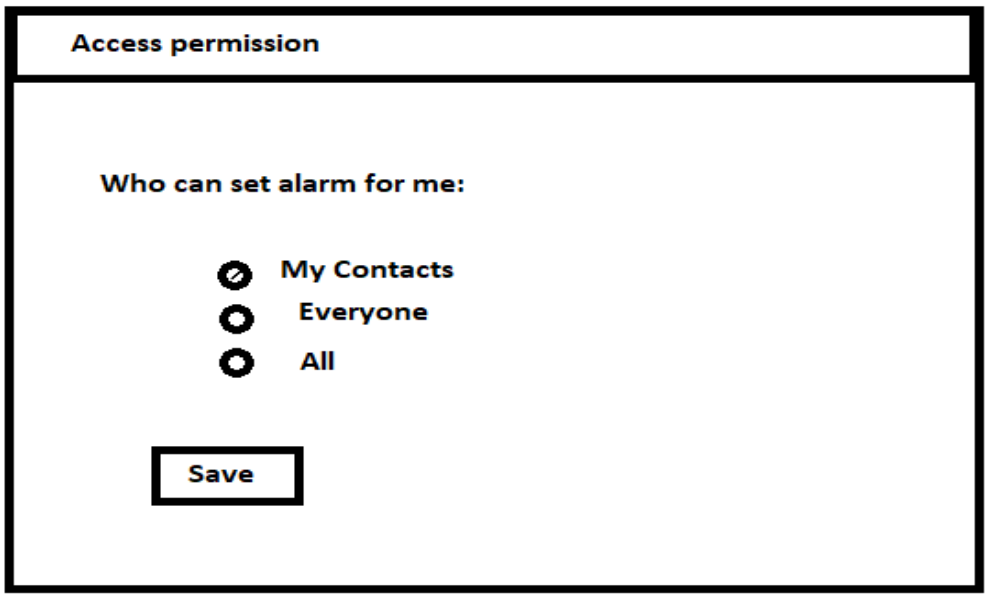
The contact search will be done in the server and load in the client. String matching algorithm is used to match the string. The backend API is called after 3 characters are entered in the contact box.



Sharing an Alarm

Access permissions:

Few permission settings are allowed to persist. The user will click on the settings button in the display. Along with the default list of settings, a setting specific to share alarm is set. In there, user can set permissions who can set alarm for them apart from him in the above explained share feature.



Access permission

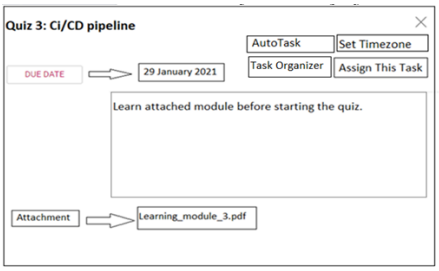
Bridging module

This is implemented to develop a solid feature combining the strengths of each utility application. In this module we have one functionality, which is kind of interaction between the calendar, clock and ToDO list module.

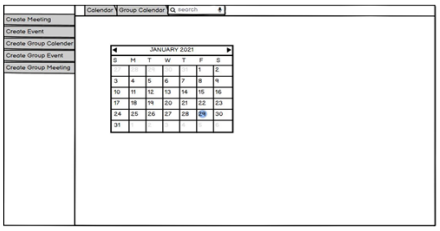
When there is a due date in ToDo, link it with the clock and calendar. In calendar as an Event and as a reminder in Clock.

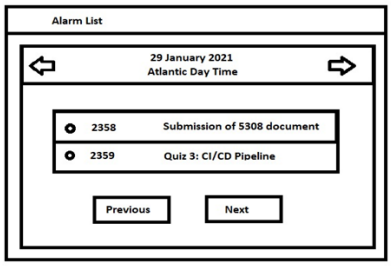
- If user will set a due date in ToDo then that should be reflected in calendar module as an event and in clock module as a reminder so that user can get the idea about his schedule while going through the calendar and clock module.
- If user wants to set a task on particular day but user don't know whether the user will be free on that day. Then by just going through the calendar module user can easily get an idea about setting the due dates accordingly.
- The changes in clock and calendar module are applicable for the group task too. In a nutshell All the member of a group receives same changes in their account.
- Here user set task which has due date of 29th Jan. So, this due date will also be reflected in calendar and clock module. (as per the below figures)

ToDo



Calendar





Clock

On doing so, the errors are handled appropriately in the backend on,

i) when the due date is passed in the input time zone then an error will be displayed over screen.

ii) When the ToDo list is a group list and the user tries to set alarm for others and if others didn't give access permission to the user an error message stating "access denied" is shown on the screen.

Agile methodology will be used to implement this project.

Tentative time line for the project completion is 4 sprints and each sprints consist of two weeks.

Team is expected to implement six stories per sprint. A scrum will be planned daily for 15 min. to track the progress.

Expected plan to meet the functional completeness:

25% - complete authentication module. Planning to decouple the database as an internal API so that anyone can use that from their individual module.

50% - each person will work on the standalone features in their application. 4 standalone features will be completed at this milestone.

75% - Two standalone features per team member and Completing complex features like, Search module inside calendar, ToDo, contact using String matching algorithm, Automatic Task Organizer and Detecting task recurrence and alarm recurrence from the title.

100 % - Implementing all the bridging modules mentioned in the above feature list and refactoring.

References:

1) Mocks are created using Microsoft Paint and https://balsamiq.cloud/