Sultan Qaboos University - College of Science
Department of Computer Science
COMP5405 : Software Patterns Project - Spring2022

Project Phase 1 : Project Description
Task Manager project

Submitted to: Dr. Yassine Al Jamoussi
Students:

Zayana Al Lamki	120042	**Coordinator**
Fatema Al Ghafri	124394	
Amna Al Nadabi	126558	

Table of Content

Introduction

Everyone can have a messy day that ends up with a non-productive day since nothing is clear and nothing is scheduled behind their eyes. Because of that, we need to have a calendar and to-do list in our life.

In our application, we suggest having a full platform that has a calendar of the year with a list that has tasks that have to be done daily or weekly, or even monthly. People who use it will have lots of features like having a group of tasks, showing the users the daily tasks on the main page, and others. These things can be written by hand on paper but by the application that we suggest, it will be easier, more effective, and predictive.

Project objectives

- To make the daily tasks done on a proper time without any delay.
- To combine the calendar with to-do lists in one app.
- To visualize the goals and priorities much better.
- To improve the memory so the user will not miss any task.
- To motivate the user to do all the tasks and mark them as done.

Motivation

Choosing this project not coming from nowhere, we choose it because we need it as students, and everyone else needs it as an employee, businessman, doctors, Parents and so on. Our application will reduce the stress in their life and get more tasks done in less time.

So there are lots of benefits that we aim to achieve by doing this project such as achieving more goals and being the successful person you want.

Requirements

Functional Requirements

1. The user will be able to create a task with duration(starting and ending time), and a title.
2. Each task can be repeated as a routine daily, weekly, monthly, yearly or specified by the user.
3. The application will allow users to create groups to gather the related tasks, and each group has a name and a description.
4. Each group has many or no tasks.
5. The application will allow the user to assign a task to at most one group.
6. The application will show the user's today's to-do list which contains the tasks specified before.
7. Each to-do list can contain tasks from different groups.
8. The application will show to the user the daily tasks on the main page and will allow the user to view all the tasks and the groups too.
9. The user can update the groups and the details of the task.
10. The user can delete the tasks and the groups.
11. When the user finishes a task he can set the task as done.

12.Each day the application will show the progress of the user's tasks accomplishments.

Nonfunctional Requirements

1. Product operations : Usability , Efficiency

   Our application has to be as easy as we can, so all of the people can use it, even the younger and older ones. It has to be a simple and very fast way to write all the tasks and assign them since it is a daily work. Our app is available for use with all operations and available 7/24.

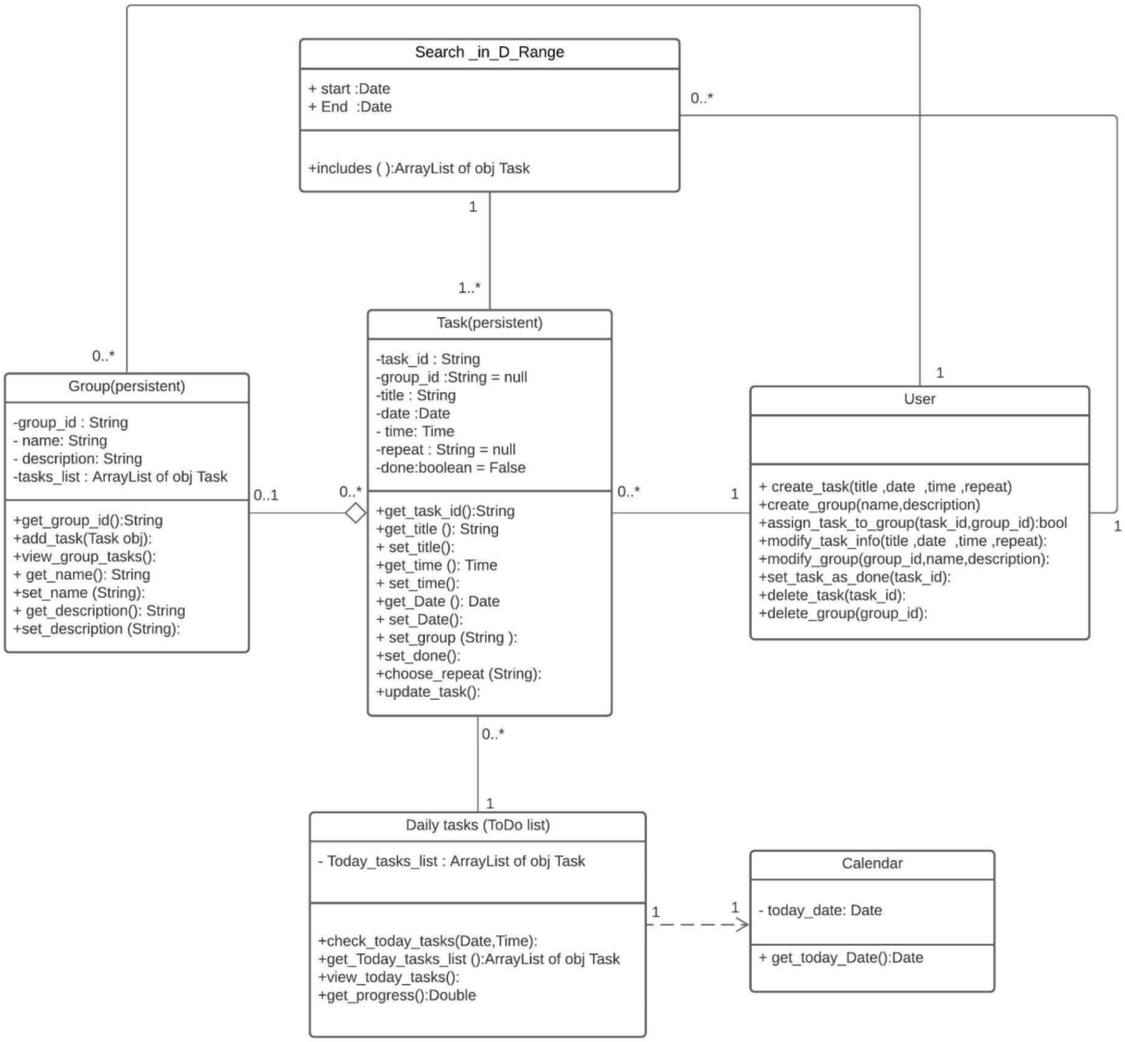
2. Product revision : Maintainability , Testability , Flexibility

   Our application will be able to be maintained for a longer period of time at a lower cost. It will also be tested to ensure that the application is of high quality and that the system operates as intended. Furthermore, our program will be tweaked to accommodate diverse situations, setups, and user expectations.

   Product transition : Portability, Reusability, Interoperability

   Our application software system will be able to be moved from its present hardware or software environment to another. Also it can interface with other systems or partner with them for event schedules , business orders and so on.

Class diagram

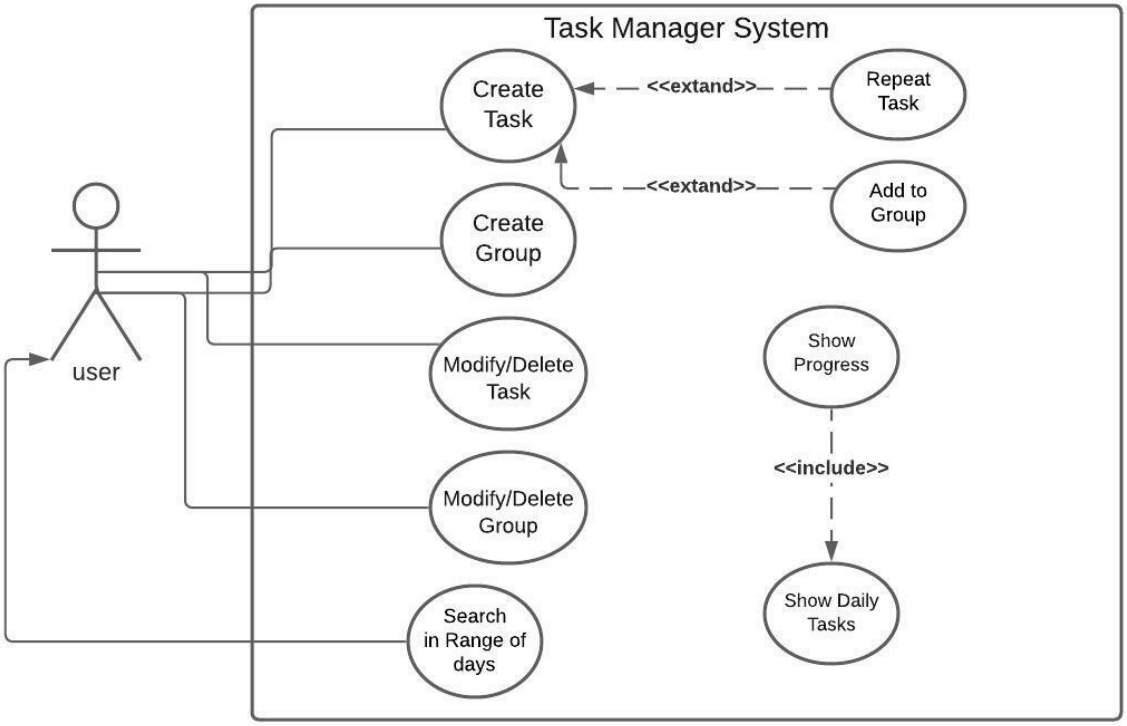


Analysis Patterns that helped us to analyze our problem

The common way to check or find out if a value is set within specified range is by checking for both start and end values compared to the tested value .Range on the other hand uses one object act as whole to decide whether a value is within the range or not, and by using the relevant operations to compare.The structure of the class is very simple. It contains two variables or containers that represent the start of the range and the end of the range. Also it uses an includes method that checks to see if the value provided is within the range or not.(*you can find additional information about this patterns in the following link:https://martinfowler.com/eaaDev/Range.html*)

In our project we are going to use the Range pattern in the “Search In Range of days” use case to give users an additional functionality to view all tasks allocated on one range. The reason we use this pattern is because we have to search in all tasks that exist to find the tasks that belong to the range, and it will be very difficult to check for the start day and the end day each time we iterate and that consumes time and memory.

Use case diagram



In this use case diagram, the system allows the user to create groups and tasks. Each task has two optional characteristics, the repeat attribute which repeats the task daily, weekly, monthly, yearly, or specified by the user. The other attribute is to add this task to an existing group. Moreover, the user can modify and delete the tasks and groups. Indeed, the system will automatically show the daily tasks for today's date, and for each task user updated as "Done" the system will show the progress of the user task accomplishments. Also, the user can search for all tasks that existed between two specified days.

Use case Descriptions

“Create Task “

Actor: User.

Summary: Create the task that needs to be done.

Description: This use case allows users to write all the tasks that they want to have on their list.

Exception: Every user is expected to have at least 1 task a day to be done.

Post-Condition: every task will be shown on the main page after creating it.

“Create Group “

Actor: User.

Summary: Create a group of tasks.

Description: This use case allows users to Group the tasks with the same category in one group which will make it easy for the user to check them together.

Exception: Every user is expected to have at least 1 category group to combine the tasks.

Post-Condition: every group will be shown in the main page with every task that includes it.

“Modify/ Delete task “

Actor: User.

Summary: Edit or delete a task that has been created before.

Description: This use case allows users to Edit the tasks as they want after they created them before or even, they can delete the task from the tasks.

Exception: Every user is expected to modify the task if something changed or delete it if the task is canceled.

Post-Condition: new tasks will be shown after the modifying and the deleting.

“Modify/ Delete Group “
Actor: User.
Summary: Edit or delete a group that has been created before.
Description: This use case allows users to Edit the group as they want after they created them before or even, they can delete the group of tasks or only delete the group and keep the tasks.
Exception: Every user is expected to modify the group if something changed or delete the group if they do not need it.
Post-Condition: new groups will be shown after the modifying and the deleting.

“Repeat task “
Actor: User.
Summary: if there is a task that wants to be done more than once.
Description: This use case allows users to have a reputation of a task daily or weekly or monthly.
Exception: User is expected to have a task that wants to be repeated more than one time.
Post-Condition: Repeated tasks will be shown on the main page every time that it is their time to be done.

“Add to group “
Actor: User.
Summary: add the task to a specified group .
Description: This use case will insert the task into one group he/she selects and the task will belong to a set of tasks referring to the same group that is done by modifying the task information .
Exception: Users are expected to find all tasks belonging to the same group in one place.
Post-Condition: the task added to the group will present in the group tasks

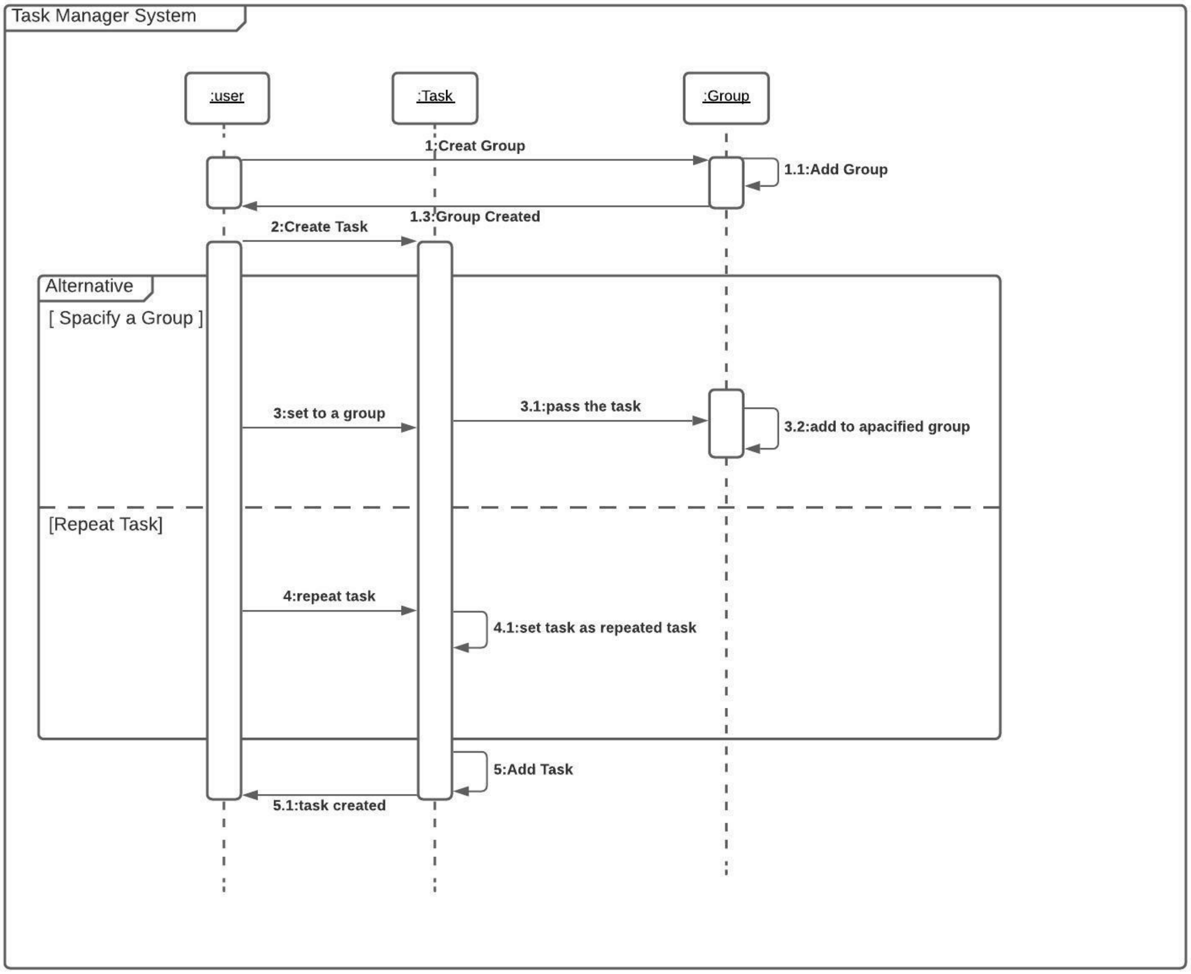
“Show Progress “
Actor: User.
Summary: show the progress of the user’s tasks accomplishments percentage on that day.
Description: This use case allows users to see all the tasks that have been done.
Exception: Every user is expected to mark the task as done after he finishes it.
Post-Condition: progress will be shown to the user on the main page.

“Show daily tasks “
Actor: User.
Summary: show the Daily tasks that must be done.
Description: This use case allows users to see all the tasks on that day.
Exception: Every user is expected to check all the tasks that he must do on that day, daily.
Post-Condition: Daily tasks will be shown to the user on the main page.

“ Search in Range of Tasks “
Actor: User.
Summary: show the tasks in a specific range of dates .
Description: This use case allows users to see all the tasks in a specific range after they entered the start and end date.
Exception: Every user is expected to enter two dates and get the tasks.
Post-Condition: Tasks will be shown to the user on the main page.

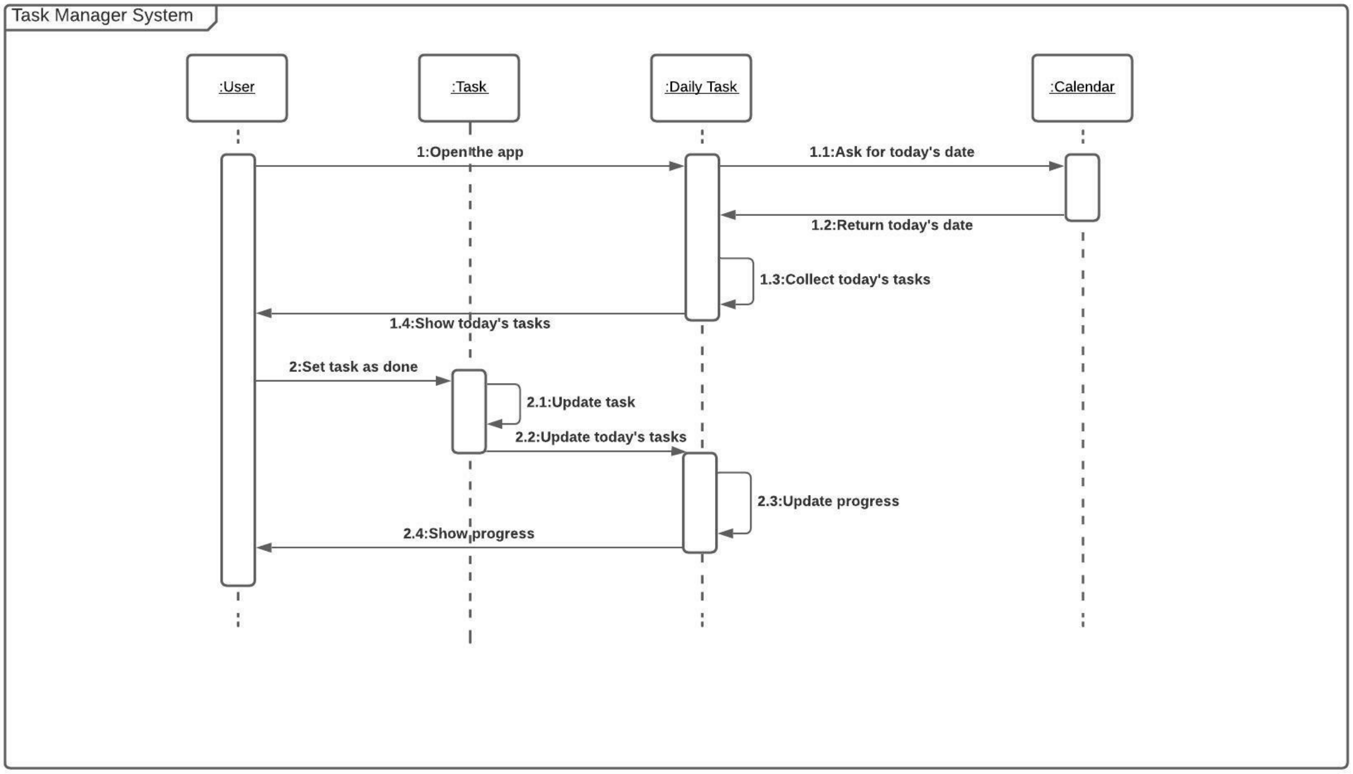
Sequence Diagrams

Sequence diagram explains (create task ,repeat task , add to group, create group)



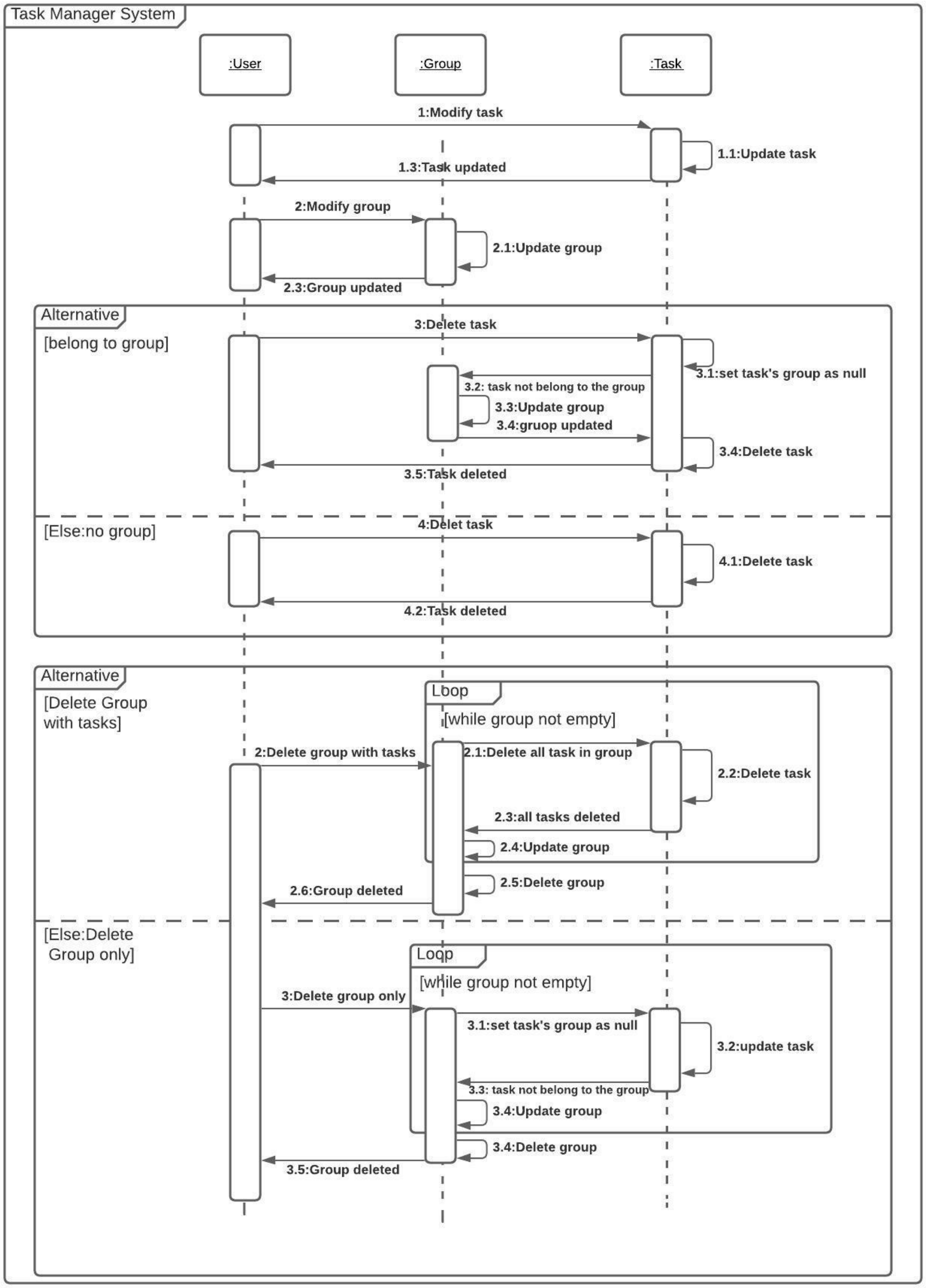
This sequence diagram represents the steps of how the system component will interact with each other when the user tries to create a group and then create a task. First, the user will create the group by requesting a new group with specified requirements then a new group will be added to the list of groups and a message will be sent to the user as feedback to confirm the creation of the new group. When the user asks to create a new task the system will check if the two optional attributes were selected, the group to which the task belongs, and the repeat attribute for the task. If the user-specified a group, the task will be passed to the group class to append it to the specified group.if the user asks to assign a task as repeated the task class will set this task as repeated to act with it differently and as required in the future. Finally, the task will be added to the list of tasks and a message will be sent as feedback for the user to verify the creation of the task.

Sequence diagram explains(show daily tasks ,show progress)



This sequence diagram explains how the system presents the daily tasks and how the progress will update when the user finishes a task. When the system is uploading and the user is opening the application, the daily task class will ask the calendar class for the date of today; so it can search and collect today's tasks and show to the user today's tasks, which is a list of tasks that need to be done on the day. If the user sets a task as done, the task will be updated and the list of today's tasks will be updated as well. This will require for the progress to be updated and the new progress will show to the user.

Sequence diagram explains(modify/delete task ,modify/delete group)

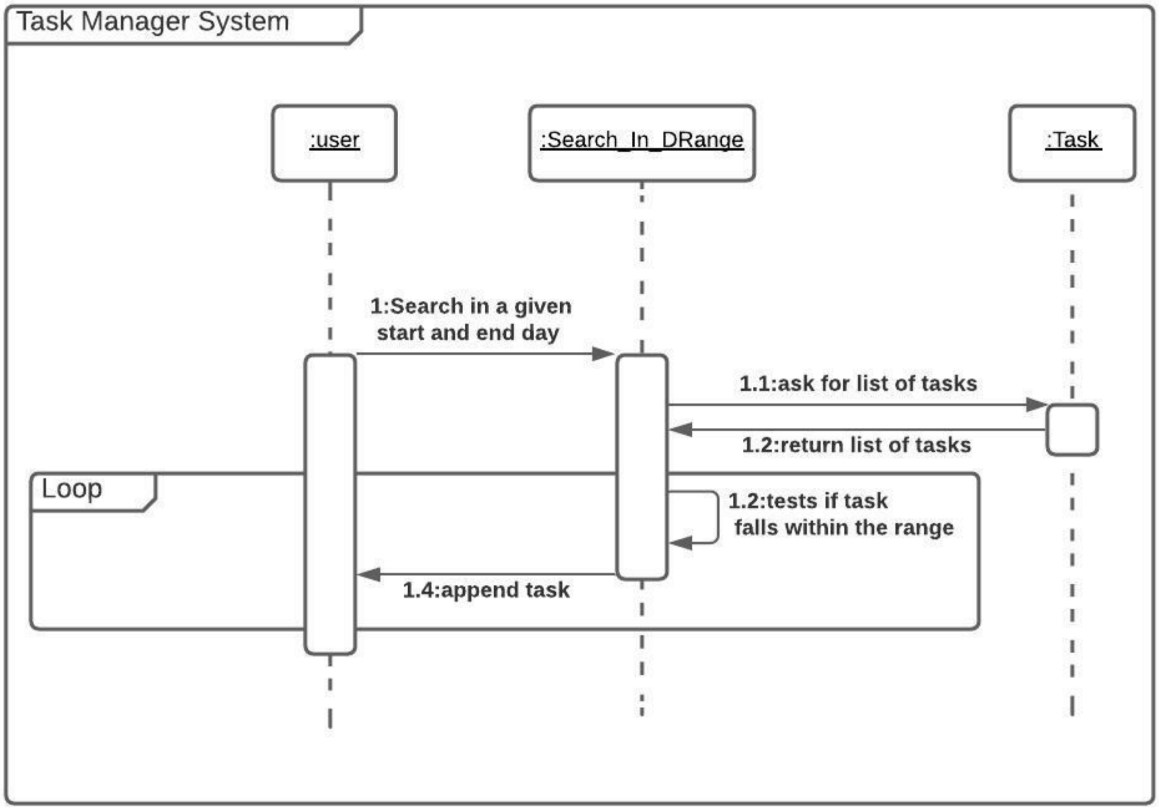


The above sequence diagram is presenting the processes of deleting and modifying Tasks and groups. The modification of a task or group is following the same procedure which is asking to modify a group or a task, the task or group will be updated, then a message will be sent to the user confirming that the group or task has been modified.

Task deletion requires checking if the task belongs to a group or not. If a task does not belong to a group then the program will delete the task then send the feedback to the user, otherwise, the system will set the task's group as null and then update the group to which it belongs, then the system will delete the task and send the feedback to the user.

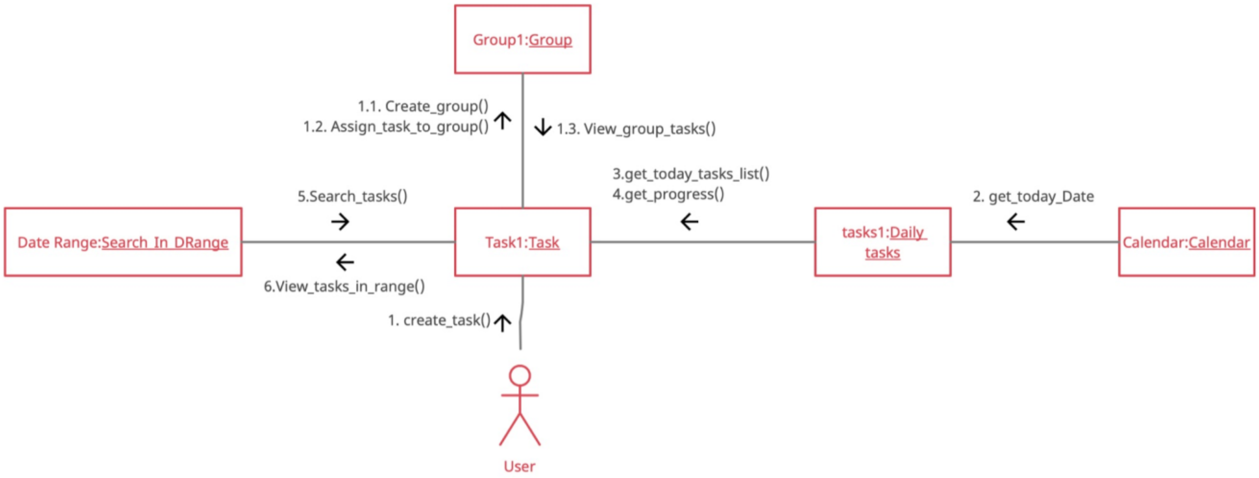
Group deletion is a more complicated process; in science, the system has to check if the user asked to delete the group only or the group with its tasks. In the case of deleting the group with its tasks, the system needs to loop through all the tasks and delete them one by one. If all tasks were deleted and the group updated and empty, then the system can delete the group. When the user asks to delete the group only, the system will loop through the task and set their group attribute to null. If all tasks were updated and the group ends with no tasks, then the system can delete the group.

Sequence diagram explains(Search in Range of days)


Task Manager System
:user
:Search_In_DRange
:Task
1:Search in a given start and end day
1.1:ask for list of tasks
1.2:return list of tasks
Loop
1.2:tests if task falls within the range
1.4:append task

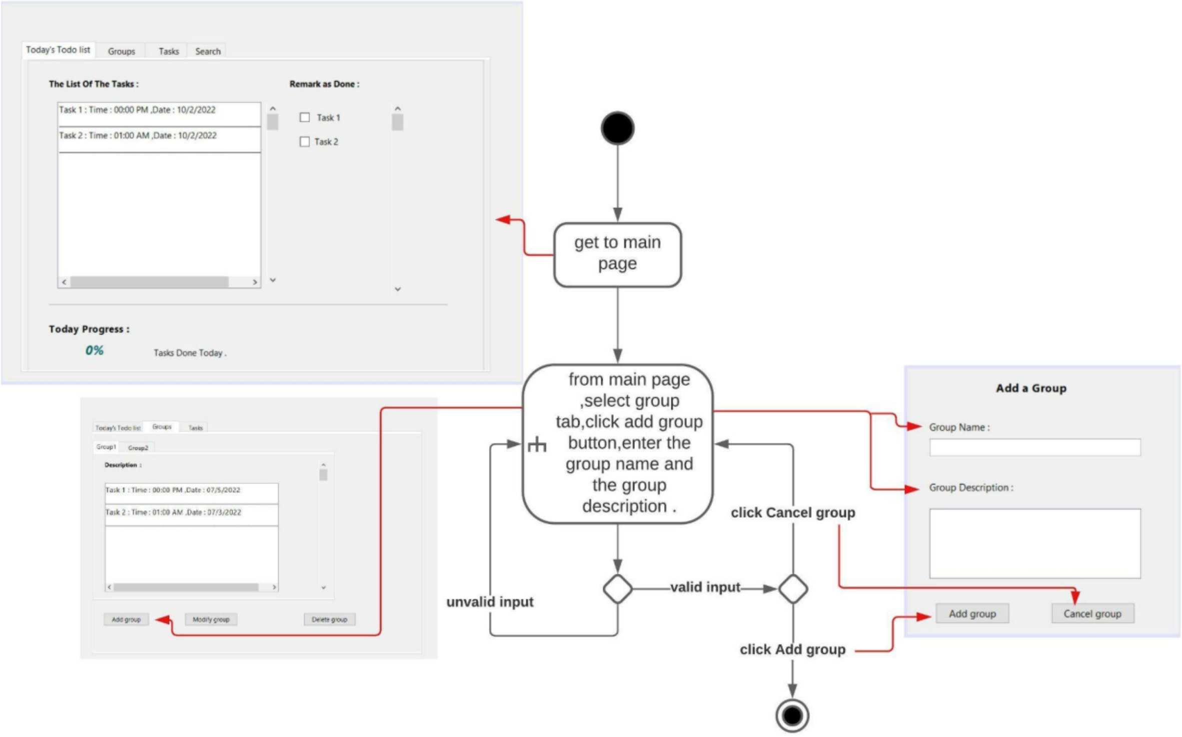
In this sequence diagram the user asks to search for all tasks that fall between two different days. The search in days range object will ask for the list of tasks to iterate through them and find the tasks that are within the range and then append or view it to the user. This loop will continue until the list of tasks finishes.

Collaboration Diagram

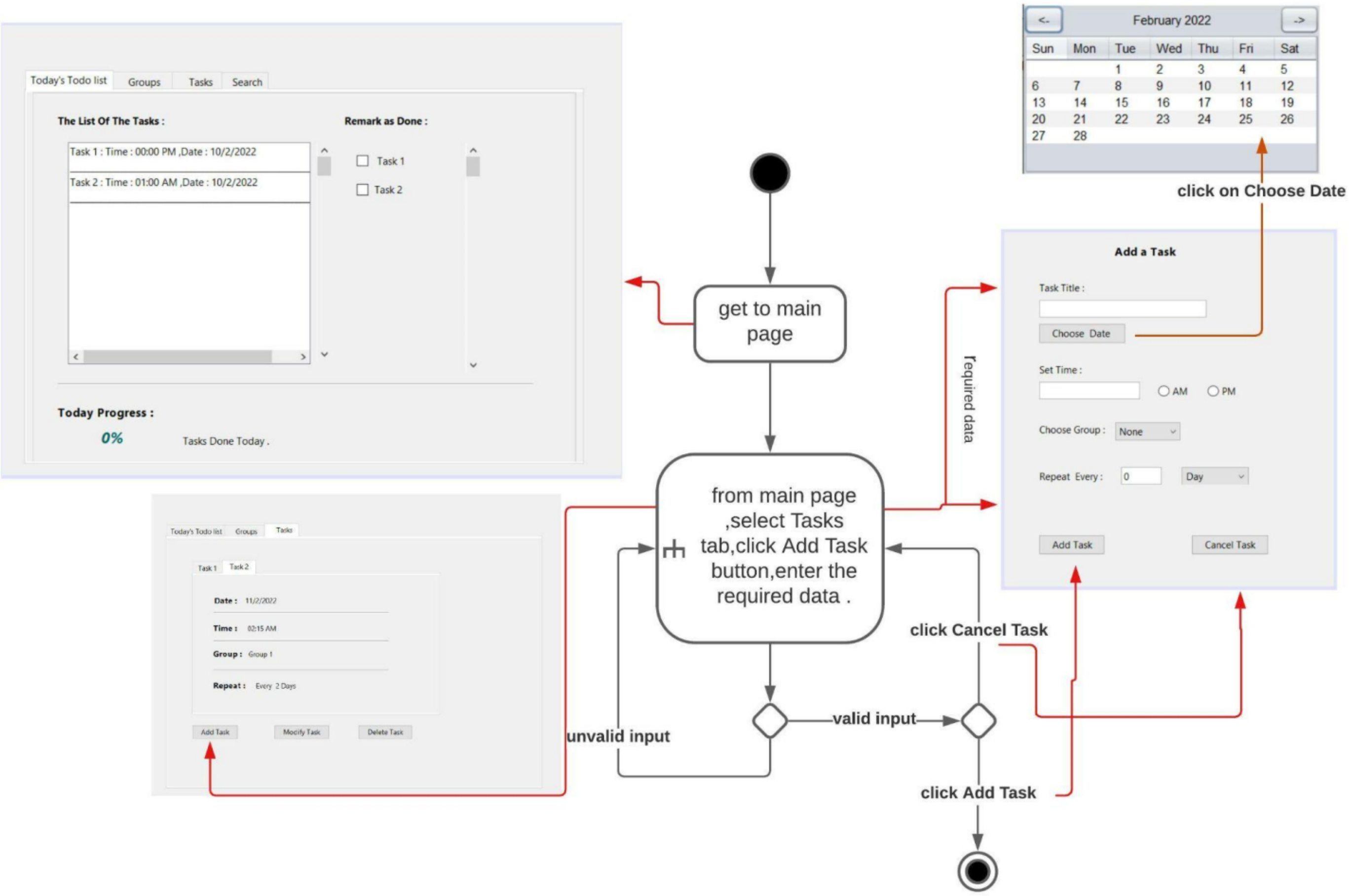


Prototype

Activity diagram for creating a group.



Activity diagram for creating a Task.



Today's Todo list | Groups | Tasks | Search

The List Of The Tasks :

Task 1 : Time : 00:00 PM ,Date : 10/2/2022

Task 2 : Time : 01:00 AM ,Date : 10/2/2022

Remark as Done :

□ Task 1

□ Task 2

Today Progress :

0% Tasks Done Today .

<- February 2022 ->

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

Add a Task

Task Title :

Choose Date

Set Time : ○ AM ○ PM

Choose Group : None

Repeat Every : 0 Day

Add Task | Cancel Task

Today's Todo list | Groups | Tasks

Task 1 | Task 2

Date : 11/2/2022

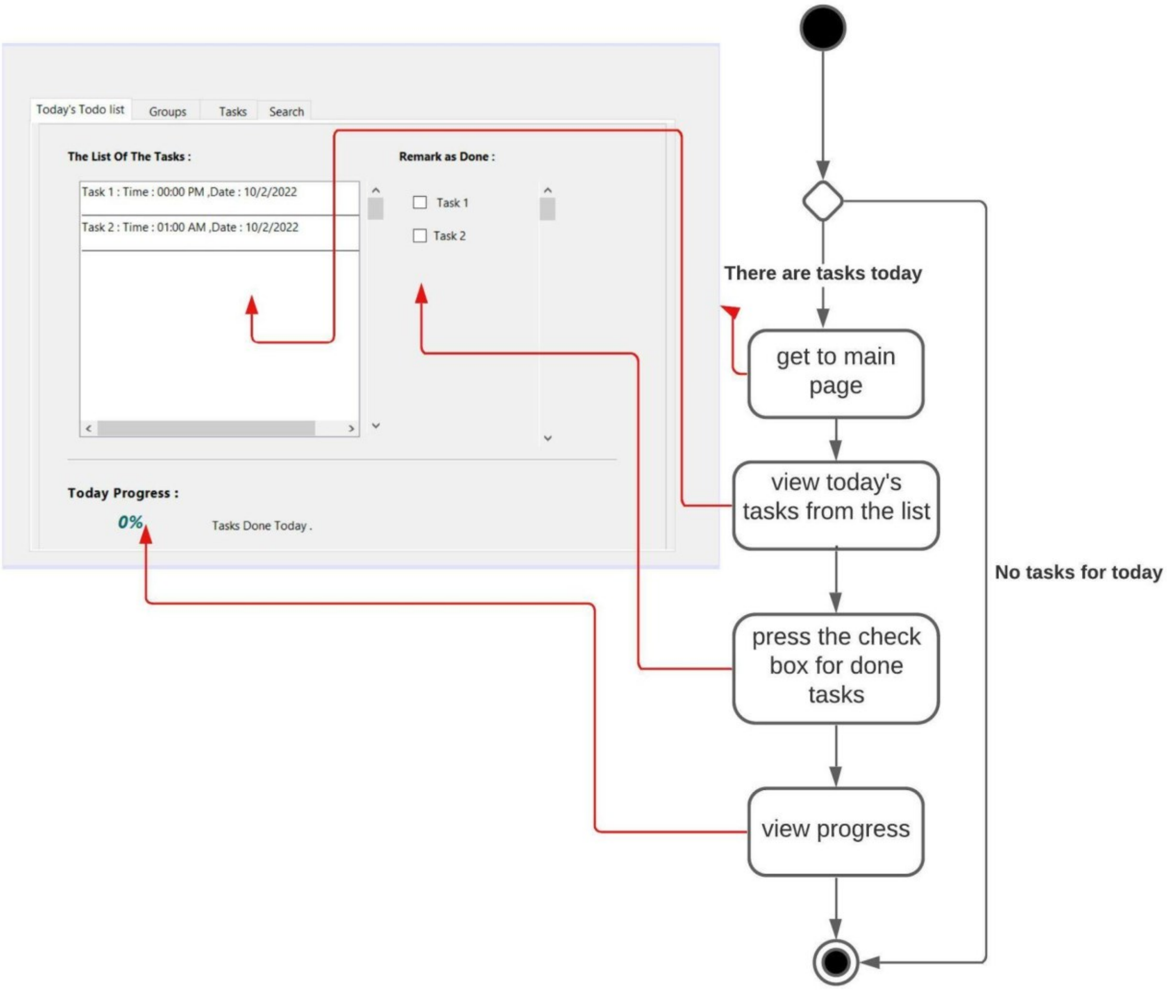
Time : 02:15 AM

Group : Group 1

Repeat : Every 2 Days

Add Task | Modify Task | Delete Task

Activity diagram for viewing daily tasks and its progress.



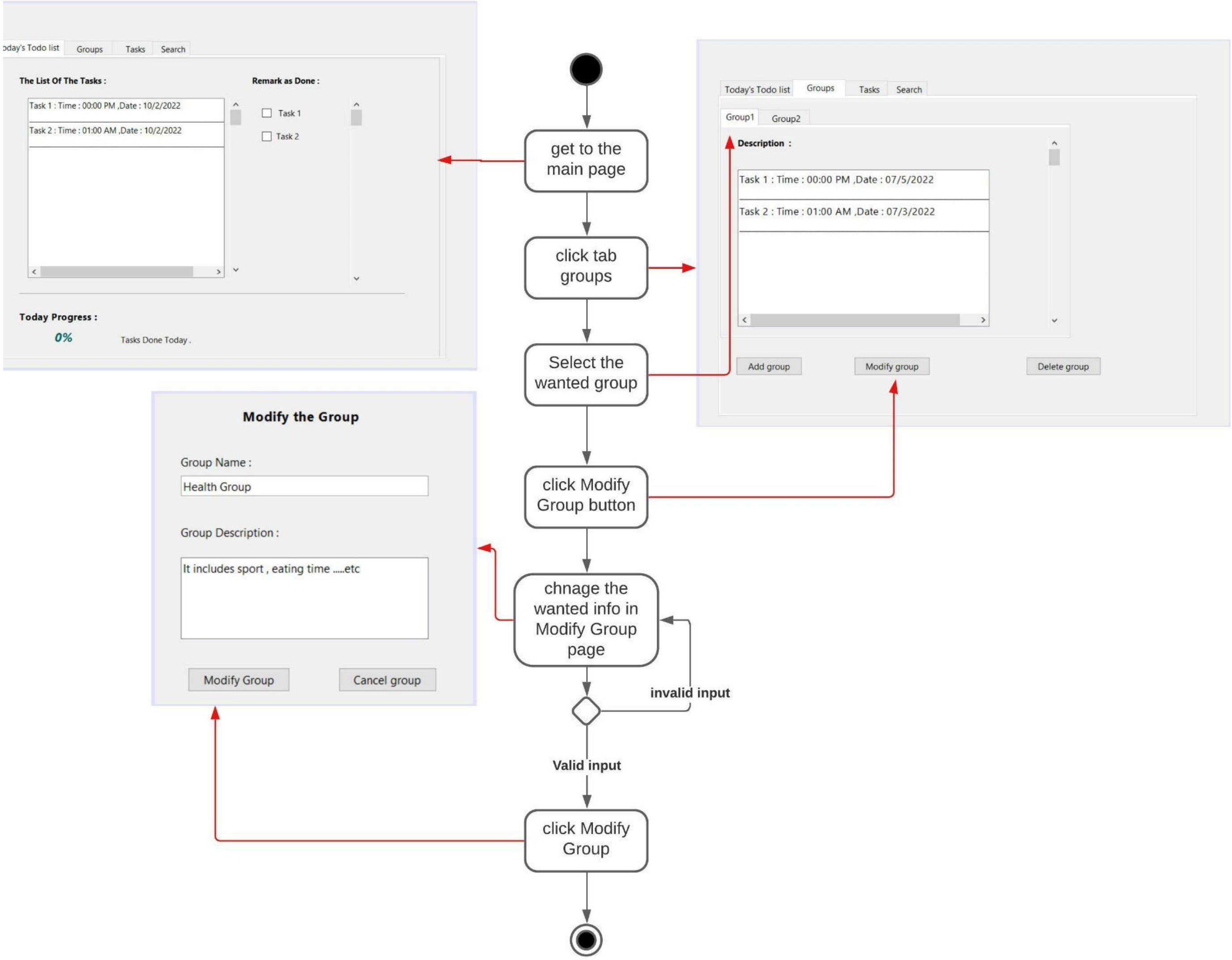
Task 1 : Time : 00:00 PM ,Date : 10/2/2022

Task 2 : Time : 01:00 AM ,Date : 10/2/2022

□ Task 1

□ Task 2

Activity diagram for modification of Group.



The List Of The Tasks :

Task 1 : Time : 00:00 PM ,Date : 10/2/2022

Task 2 : Time : 01:00 AM ,Date : 10/2/2022

Remark as Done :

□ Task 1

□ Task 2

Today Progress :

0%   Tasks Done Today .

Description :

Task 1 : Time : 00:00 PM ,Date : 07/5/2022

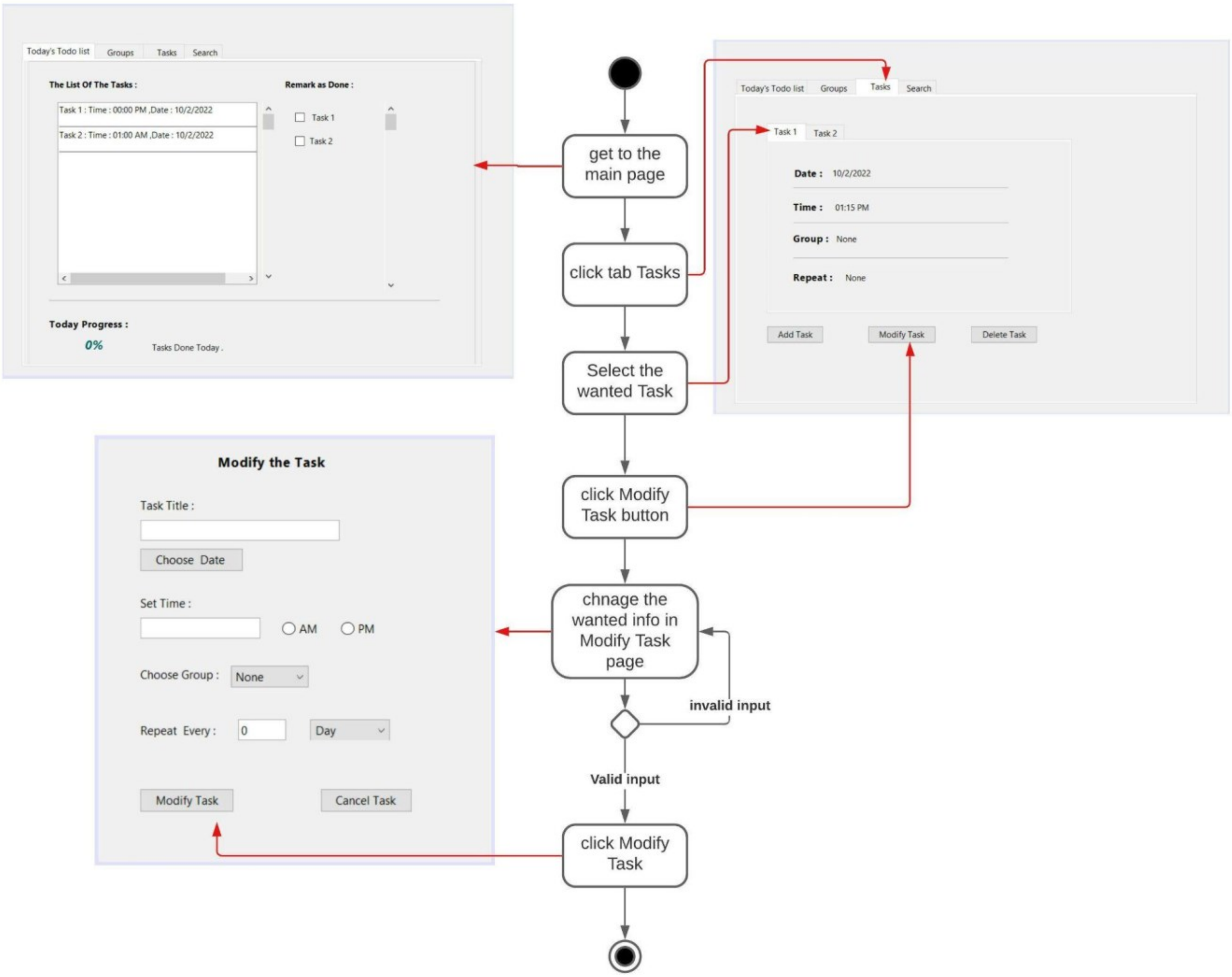
Task 2 : Time : 01:00 AM ,Date : 07/3/2022

Modify the Group

Group Name : Health Group

Group Description : It includes sport , eating time .....etc

Activity diagram for modification of Task.



The List Of The Tasks :

Task 1 : Time : 00:00 PM ,Date : 10/2/2022

Task 2 : Time : 01:00 AM ,Date : 10/2/2022

Remark as Done :

□ Task 1

□ Task 2

Today Progress :

0% Tasks Done Today .

Date : 10/2/2022

Time : 01:15 PM

Group : None

Repeat : None

Modify the Task

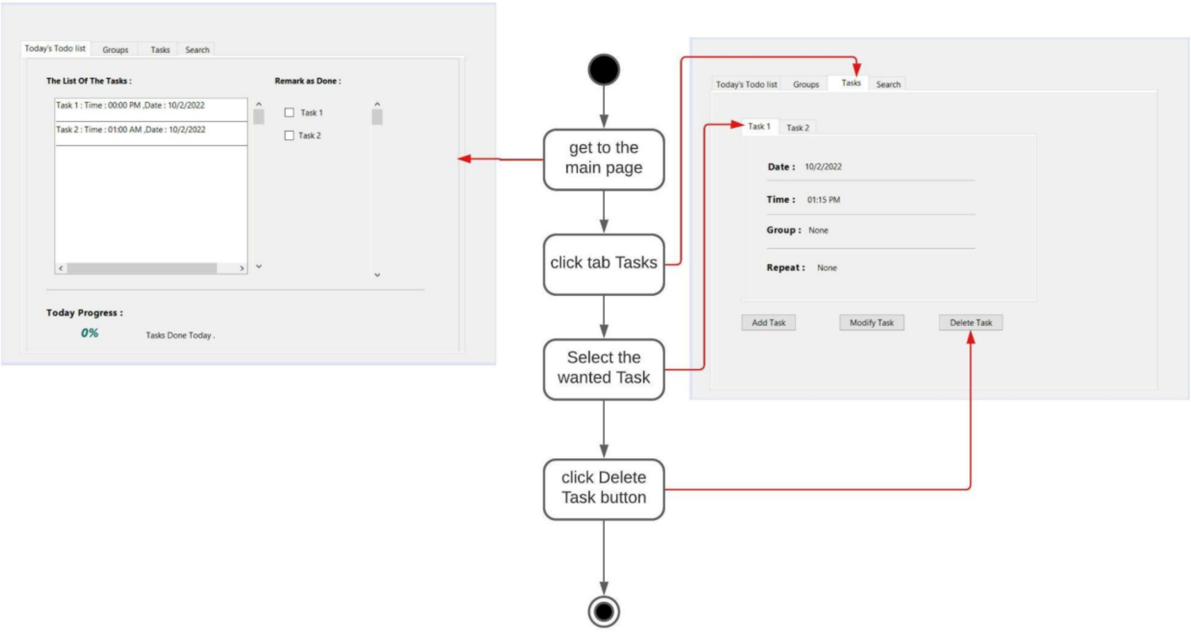
Task Title :

Set Time : ○ AM ○ PM

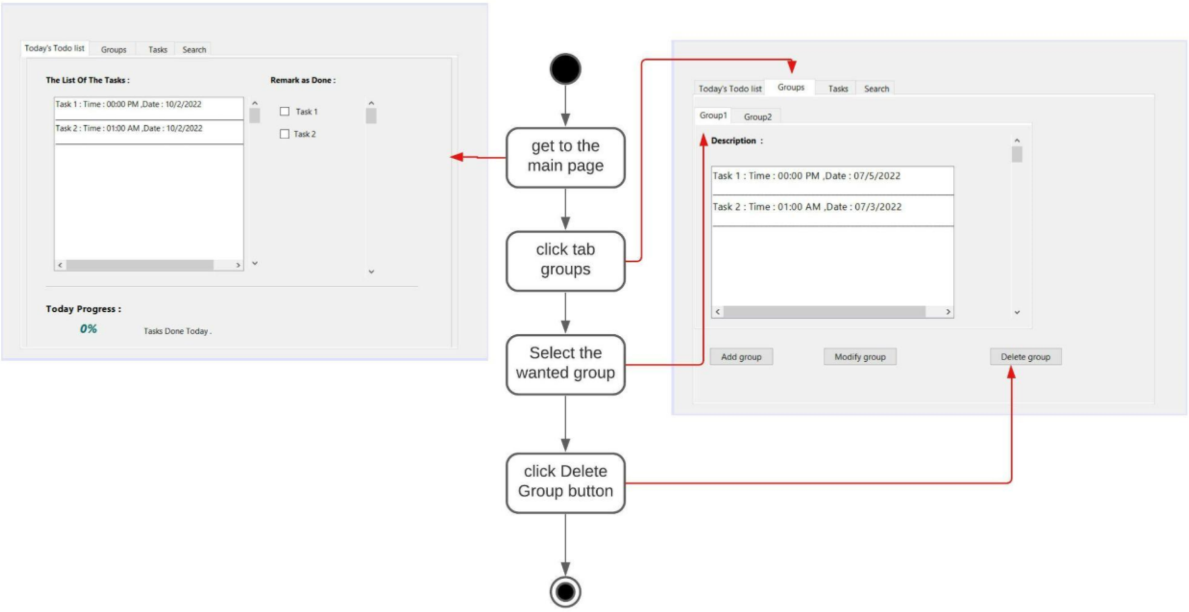
Choose Group : None

Repeat Every : 0 Day

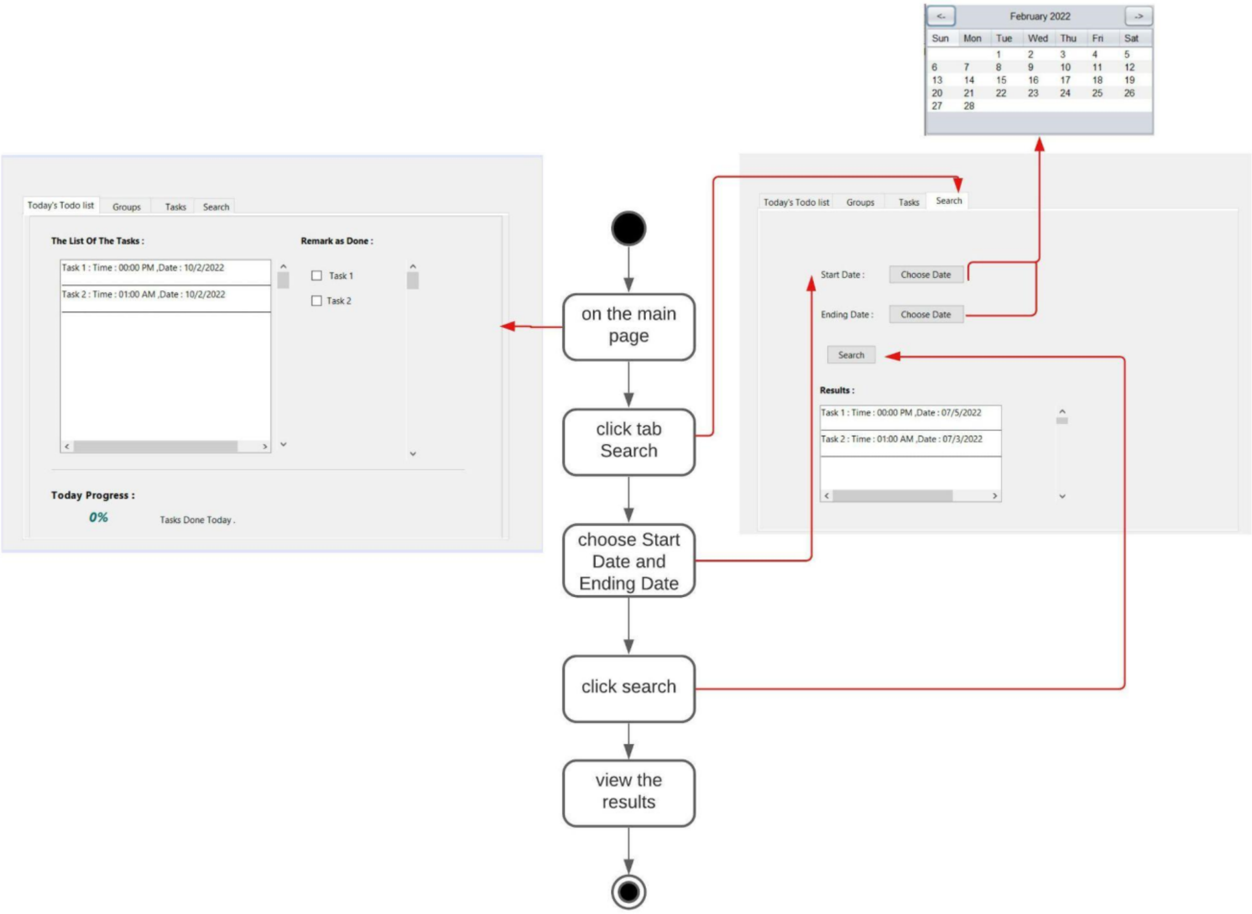
Activity diagram for deleting a task.



Activity diagram for deleting a group.



Activity diagram for Search in Range of Tasks



The List Of The Tasks :

Task 1 : Time : 00:00 PM ,Date : 10/2/2022

Task 2 : Time : 01:00 AM ,Date : 10/2/2022

Remark as Done :

□ Task 1

□ Task 2

Today Progress :

0% Tasks Done Today .

February 2022

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

Start Date :

Ending Date :

Results :

Task 1 : Time : 00:00 PM ,Date : 07/5/2022

Task 2 : Time : 01:00 AM ,Date : 07/3/2022

Contribution

Team member	Contribution
Zayana Al Lamki	Sequence diagrams, analysis patterns , use case diagrams ,functional requirements .
Fatma Al Ghafri	Description of the use cases ,introduction , motivation , objectives , non-functional requirements ,collaboration diagram .
Amna Al Nadabi	Prototype , activity diagrams , functional requirements , class diagram .