TECHNICALITIES OF TASK LISTING APP

# APP MODEL & FLOW:

**task-home**

**my-dialog**

**addTask()**

**task-list**

**getTask()**

**xyz()** **= Service function**

**= Component**

Figure: App model & flow

Note:

1. Arrows depict navigation which is controlled by routing functionality.
2. task-list is a child component of task-home.
3. The role of an additional component called PageNotFoundComponent comes into picture when an invalid URL is entered in the address bar. It is a simple notification to the user stating the page they requested wasn’t found.

All the components depend on the TaskModel interface. It has the following properties:

1. id: number
2. taskName: string
3. taskDescription: string
4. taskDueDate: Date

## Role of Service:

The components communicate and pass data between each other through a service called TaskService. This service provides the following functions:

getTask(): This function returns the list of all tasks added. It basically returns an array of type TaskModel, so it can store several task details.

addTask(task: TaskModel): Called from the MyDialogComponent component, this function creates a new task entry in a central array. It is from this array that all other functions pick up details.

## Role of Forms:

Notice the use of reactive driven form and initialization of form values using form builder in MyDialogComponent component.

## Role of Custom Directive:

Notice the task table rows highlighted by different colors on mouse hover, depending on the task’s due/completion date. This feature is controlled by the MyMouseOverDirective custom directive. The background color is determined using switch-case as follows:

1. If Due date = Current Date, the background color is GREEN
2. If Due date < Current Date, the background color is RED
3. If Due date > Current Date, the background color is YELLOW

|  |
| --- |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Role of MatDialog:

The implementation of the dialog window that takes input values for a task is done using custom modal window that uses Angular’s Material Dialog or **MatDialog** services.