



## Implementing an Automated Timesheet Management System in Salesforce technology

Kiran Konakalla

Kiran.Konakalla7@gmail.com

---

### ABSTRACT

In modern organizations, managing employee timesheets, particularly for full-time employees and contractors, is a critical function for HR departments. This paper discusses the implementation of a timesheet management system within Salesforce, using custom objects, formula fields, Lightning Web Components (LWC), and Apex code in salesforce. The proposed system tracks the hours worked by employees, aggregates them by week and month, and provides an approval workflow for managers. This Salesforce-based solution enhances efficiency, transparency, and accuracy in timesheet management.

**Keywords:** Salesforce, Timesheet Management, Custom Objects, Lightning Web Components, Apex, Human Resources, Approval Workflow

---

### INTRODUCTION

It is very important to accomplish cost reduction in product design; every company wants to remain competitive in the Managing employee and contractor timesheets can be a cumbersome process, particularly when there are different categories of employees, such as full-time workers, contractors and interns. Traditional methods often rely on external tools, spreadsheets, or manual tracking, which are prone to errors and inefficiencies. This paper explores how Salesforce can be used to streamline this process by creating a custom timesheet object, automating approval workflows, and leveraging Lightning Web Components for enhanced user experience.

### MAIN BODY

#### Problem Statement:

Organizations face several challenges in managing timesheets:

1. Tracking the number of hours worked by full-time employees and contractors across multiple projects or tasks.
2. Submitting timesheets for approval by managers.
3. Aggregating hours weekly and monthly for payroll and reporting purposes.
4. Providing a system that is easy to use and integrate within existing Salesforce infrastructure.

#### Solution:

##### Custom Object: Timesheets\_\_c

To address these challenges, a custom object called `Timesheets__c` will be created in Salesforce. This object will store information related to the time worked by employees, which will be linked to their user or contact record.

Fields on the `Timesheets__c` object:

- **Name\_\_c:** A lookup field to `User` or `Contact` to store the name of the person submitting the timesheet.
- **Date\_\_c:** A date field to track the specific date for the timesheet entry.
- **HoursWorked\_\_c:** A number field to record the hours worked on that day.
- **WeekAggregate\_\_c:** A formula field that calculates the total hours worked for the week.
- **MonthAggregate\_\_c:** A formula field that calculates the total hours worked for the month.
- **ApprovalStatus\_\_c:** A picklist field that tracks the approval status of the timesheet, with values such as 'Submitted', 'Approved', and 'Rejected'.

**Design Approaches:****1. Option 1: One Entry per Day**

○ Employees would log hours worked each day by creating a new timesheet record. A Lightning Web Component (LWC) could be used to simplify the user interface by showing a streamlined form that auto-fills the date and allows employees to input their hours.

**2. Option 2: One Entry per Week**

○ The alternative approach involves allowing employees to enter their hours for the entire week at once. In this case, the `Timesheets__c` object would have fields for Monday to Sunday, and the employee would input their hours for each day on a single screen. The record would then calculate the total weekly hours and route the timesheet to the manager for approval.

**LWC Component:**

To improve the user interface, an LWC would display a simplified view of the timesheet. The component would allow the user to input hours for each day of the current week, providing an easy-to-use interface that displays the current week's dates dynamically.

**LWC Code Snippet:**

```
import { LightningElement, track, wire } from 'lwc';
import { getRecord } from 'lightning/uiRecordApi';
import submitTimesheet from '@salesforce/apex/TimesheetController.submitTimesheet';

export default class TimesheetComponent extends LightningElement {
  @track mondayHours = 0;
  @track tuesdayHours = 0;
  @track wednesdayHours = 0;
  @track thursdayHours = 0;
  @track fridayHours = 0;
  @track saturdayHours = 0;
  @track sundayHours = 0;

  get totalHours() {
    return this.mondayHours + this.tuesdayHours + this.wednesdayHours +
      this.thursdayHours + this.fridayHours + this.saturdayHours +
this.sundayHours;
  }

  handleHoursChange(event) {
    const field = event.target.name;
    const value = parseFloat(event.target.value);
    if (field === 'monday') this.mondayHours = value;
    if (field === 'tuesday') this.tuesdayHours = value;
    if (field === 'wednesday') this.wednesdayHours = value;
    if (field === 'thursday') this.thursdayHours = value;
    if (field === 'friday') this.fridayHours = value;
    if (field === 'saturday') this.saturdayHours = value;
    if (field === 'sunday') this.sundayHours = value;
  }

  handleSubmit() {
    submitTimesheet({
      monday: this.mondayHours,
      tuesday: this.tuesdayHours,
      wednesday: this.wednesdayHours,
      thursday: this.thursdayHours,
      friday: this.fridayHours,
      saturday: this.saturdayHours,
      sunday: this.sundayHours,
      total: this.totalHours
    })
  }
}
```

```

        .then(() => {
            // Handle success
        })
        .catch((error) => {
            // Handle error
        });
    }
}

```

**Apex Controller:**

The Apex class would handle the logic for submitting the timesheet and routing it for approval based on the total hours worked and other criteria.

**Apex Code Snippet:**

```

public class TimesheetController {
    @AuraEnabled
    public static void submitTimesheet(Decimal monday, Decimal tuesday, Decimal
wednesday,
                                     Decimal thursday, Decimal friday, Decimal
saturday, Decimal sunday,
                                     Decimal total) {
        Timesheets__c ts = new Timesheets__c();
        ts.Monday__c = monday;
        ts.Tuesday__c = tuesday;
        ts.Wednesday__c = wednesday;
        ts.Thursday__c = thursday;
        ts.Friday__c = friday;
        ts.Saturday__c = saturday;
        ts.Sunday__c = sunday;
        ts.TotalHours__c = total;
        ts.ApprovalStatus__c = 'Submitted';
        insert ts;

        // Optional: Trigger approval process if needed
    }
}

```

**Approval Workflow:**

Salesforce provides an out of the box approval process that can be used for timesheet submissions. Once an employee submits a timesheet, it will be routed to their manager for approval. If the timesheet exceeds a certain number of hours (e.g., 40 hours for full-time employees), the approval process can be customized to include multiple levels of approval. Also we can send the time sheets for approval whenever an employee is requesting time off.

**Configuration Changes:**

1. Create a custom object: `Timesheets__c`
2. Add fields: `Name__c`, `Date__c`, `HoursWorked__c`, `WeekAggregate__c`, `MonthAggregate__c`, `ApprovalStatus__c`
3. Set up the approval process for timesheet submission.
4. Create an LWC for the timesheet input. This can be optional. We can also use the out of the box salesforce UI page for the input values.
5. Create Apex classes to handle the business logic and submission.

**IMPACT AND SCOPE**

The implementation of timesheet management in Salesforce provides significant benefits to HR departments. It simplifies the tracking of hours for both full-time employees and contractors, improves accuracy in payroll calculations, and provides a transparent and automated approval process for managers. Additionally, with the use of Lightning Web Components and Apex, the solution is scalable and can be extended to meet future business needs. For organizations with a mobile workforce, the Salesforce mobile app combined with this timesheet solution can empower employees to log their hours from anywhere, ensuring that time tracking is not delayed.

### CONCLUSION

By implementing timesheet management within Salesforce, organizations can streamline HR operations, reduce manual processes, and increase transparency and accountability. The use of custom objects, LWC, Apex, and Salesforce's approval processes makes it a powerful tool for tracking employee and contractor hours, ultimately improving the overall efficiency of HR departments.

### REFERENCES

- [1]. Salesforce Developer Documentation. "Custom Objects in Salesforce." Available: <https://developer.salesforce.com/>.
- [2]. Salesforce. "Approval Processes: Automate Your Approval Requests." Available: [https://help.salesforce.com/articleView?id=approval\\_define.htm](https://help.salesforce.com/articleView?id=approval_define.htm).
- [3]. Williams, A. "Using Apex to Automate HR Processes in Salesforce." Cloud Technology Journal, vol. 8, no. 4, pp. 45-56, 2019. Available: <https://cloudtechjournal.com>.
- [4]. Salesforce.com. "Lightning Web Components Overview." Available: <https://developer.salesforce.com/docs/component-library/overview/components>.
- [5]. Salesforce.com. "Process Builder Overview." Available: [https://help.salesforce.com/articleView?id=process\\_overview.htm](https://help.salesforce.com/articleView?id=process_overview.htm).
- [6]. Brown, L. "Timesheet Management Using Apex and Custom Objects." CRM Developer Journal, vol. 9, no. 2, pp. 102-118, 2019. Available: <https://crmdeveloperjournal.com>.