**Smart Attendance and Access Management System (SAAMS)**

**Shift-View Component Documentation**

Revision History:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Prepared / Edited By** | **Changes** |
| 0.1 | 14/03/2025 | Vasanth | Initial Draft |

**Introduction**

**Purpose**

This document describes the **Shift View Component**, which provides a user interface for viewing, adding, updating, and deleting shifts in an Electron-based application.

**Scope**

* Covers the component’s **functionality**, **UI elements**, **data structure**, and **integration** with Electron APIs.
* Does not cover backend implementation or the **Shift model** definition.

**Overview**

The **Shift View Component** renders a **data grid** displaying shift details and includes buttons for:

* **Refreshing**
* **Adding**
* **Modifying**
* **Deleting**

Dialogs (modals) handle input for adding and updating shifts, with validation rules ensuring data consistency.

**UI Elements and Functionalities**

**1. Refresh Button**

* **Label:** "Refresh" (with a <Refresh /> icon)
* **Use:** Reloads the list of shifts displayed in the data grid.

**2. Add Button**

* **Label:** "Add" (with an <Add /> icon)
* **Use:** Opens a modal to create a new shift.
* **Modal Fields:**
  1. **Shift Name**
     + **Label:** "Shift Name"
     + **Type:** Text
  2. **Shift Code**
     + **Label:** "Shift Code"
     + **Type:** Text (Min: 2 characters, Max: 4 characters)
     + **Validation:** Should not contain spaces.
  3. **Shift Type**
     + **Label:** "Shift Type"
     + **Type:** Dropdown (ROTATIONAL or NON\_ROTATIONAL)
  4. **Timing Fields**
     + Entry Time
     + Entry Grace Time
     + Exit Lunch
     + Entry Lunch
     + Exit Time
     + Exit Grace Time
     + Overtime Allowance
* **Cancel Button:** Closes the modal without saving.
* **Add Button:** Submits the form to create a new shift.

**3. Modify Button**

* **Label:** "Modify"
* **Use:** Opens a modal to edit a single selected shift.
* **Validations:**
  + If **multiple shifts** are selected, it shows a message: "Select only one item to edit."
  + If **no shift** is selected, it shows a message: "Select an item to edit."
  + Otherwise, it allows editing.

**4. Delete Button**

* **Label:** "Delete"
* **Use:** Deletes one or more selected shifts.
* **Validations:**
  + If no shift is selected, it shows "Select an item to delete."
  + If deletion fails, it displays an error message.

**Use Cases**

**1. Viewing Existing Shifts**

**Description**

Users can view a list of all configured shifts in a **tabular format**.

**Preconditions**

* The application is running.
* Shifts have been previously created and stored in the backend.

**Steps**

1. The user opens the application.
2. The shifts are displayed in a **DataGrid** with columns:
   * ID
   * Name
   * Code
   * Type
   * Entry Time
   * Entry Grace Time
   * Exit Lunch
   * Entry Lunch
   * Exit Time
   * Exit Grace Time
   * Overtime Allowance
3. Users can scroll through the list or use pagination (5 rows per page).

**Postconditions**

* The user sees an **up-to-date list** of shifts.

**Exceptions**

* If the API call fails, an **error modal** appears: "Error fetching data."

**2. Refreshing the Shift List**

**Description**

Users can manually refresh the shift list to ensure it reflects the **latest data** from the backend.

**Preconditions**

* The application is running.

**Steps**

1. The user clicks the "Refresh" button.
2. The **DataGrid** updates with the latest shifts.

**Postconditions**

* The shift list reflects the current state of the backend.

**Exceptions**

* If the API fails, an error modal displays: "Error fetching data."

**3. Adding a New Shift**

**Description**

Users can add a **new shift** with specific details.

**Preconditions**

* The application is running.

**Steps**

1. The user clicks the "Add" button.
2. A **modal** opens with fields for:
   * **Shift Name** (text)
   * **Shift Code** (text, 2-4 characters, no spaces)
   * **Shift Type** (dropdown: ROTATIONAL or NON\_ROTATIONAL)
   * **Entry and Exit times** (Time Picker)
3. The user fills in the required fields and clicks "Add."
4. The form **validates**:
   * Code should not contain spaces.
   * Shift times should follow the format D.HH:mm:ss
5. A success or error modal appears based on the API response.
6. The **shift list refreshes automatically.**

**Postconditions**

* A new shift is added to the backend and displayed in the UI.

**Exceptions**

* **Invalid shift code** triggers an error message: "Code should not contain spaces."
* **API failure** shows an error modal.

**4. Updating an Existing Shift**

**Description**

Users can modify the details of an **existing shift**.

**Preconditions**

* The application is running.
* At least one shift exists and is selected.

**Steps**

1. The user selects **one** shift from the **DataGrid**.
2. The user clicks the "Modify" button.
3. A **modal** opens, pre-filled with the selected shift’s details.
4. The user edits the required fields.
5. The user clicks "Update."
6. The form validates the input.
7. A success or error modal appears, and the shift list refreshes.

**Postconditions**

* The selected shift is updated in the backend and UI.

**Exceptions**

* Selecting **multiple** shifts triggers an error modal: "Select only one item to edit."
* Selecting **no shift** triggers an error modal: "Select an item to edit."
* **Validation or API errors** are displayed in modals.

**5. Deleting a Shift**

**Description**

Users can **remove** one or more existing shifts.

**Preconditions**

* The application is running.
* At least one shift exists and is selected.

**Steps**

1. The user selects **one or more** shifts from the **DataGrid**.
2. The user clicks the "Delete" button.
3. The shift list refreshes.

**Postconditions**

* The selected shifts are **removed** from the backend and UI.

**Exceptions**

* If **no shift** is selected, a modal displays: "Select an item to delete."
* If any deletion **fails**, an error modal shows the failed ID.