# Use Case: Sales Entry

**Primary Actor**: Sales Person/ Admin

**Description**: Record sales transactions, update stock levels, and ensure accurate inventory management.

## Main Success Scenario:

1. The user navigates to the **Sales Page**.
2. The system automatically fetches the available price items from the **Product Price Table**, including productId, productSku, PricePerUnit, and UnitOfMeasure.
3. The user begins a new sale by entering:
   * The **Product Name** (which maps to the product details).
   * The **Units** to be sold.
   * The **Amount** (total price for the sale).
4. The system performs the following validations:
   * The system checks if the entered **Units** match a valid PricePerUnit and UnitOfMeasure combination for that product from the **Product Price Table**. If no match is found, an error is returned indicating that the unit does not match any valid pricing for the product.
   * The system fetches the current **Stock Level** of the selected product from the **Stock Table**. If the requested **Units** exceed the available stock, an error is returned, notifying the user that the stock is insufficient.
5. If any of the above errors occur (invalid units, stock level exceeded, or missing inputs), the user is prevented from submitting the sale. The form cannot be submitted until the errors are corrected.
6. If all inputs are valid and no errors are present:
   * The user clicks **Submit**.
   * The system records the sale in the **Sales Table**, storing details such as productId, Units Sold, Total Amount, and dateCreated.
   * The system updates the **Stock Table**, reducing the stock quantity of the sold product by the number of units sold.
7. After updating the stock:
   * The system checks if the current stock level for the product is less than or equal to its ReorderLevel (fetched from the **Product Price Table**).
   * If the stock level is below or at the reorder level, the system:
     + Sends a **Notification** to the user, warning that the product has reached its reorder level.
     + Automatically sends a message to the admin notifying them about the low stock level and the need for restocking.

## Extensions:

* **Price Mismatch Error**: If the user inputs units that don't match any valid PricePerUnit and UnitOfMeasure from the **Product Price Table**, the system returns an error and highlights the mismatch.
* **Stock Insufficiency Error**: If the user inputs more units than are available in stock, the system returns an error indicating that the stock is insufficient and prevents form submission.
* **Reorder Level Notification**: If, after the sale, the total stock falls below or is equal to the ReorderLevel, the system automatically triggers a notification and an admin message.

## Alternative Flows:

* **Invalid Product or Missing Data**: If the user fails to select a product, enter units, or provide the amount, the system prevents form submission and returns an error message asking for the missing input.

## Postconditions:

* The sale is successfully recorded in the **Sales Table**.
* The **Stock Table** is updated to reflect the reduction in stock due to the sale.
* If the stock falls to or below the ReorderLevel, a notification is sent to the user and a message is sent to the admin for stock replenishment.