

MADE BY-Prajjwal Pandey Empld-46022672 268prajjwal@gmail.com

Abstract:

The Order Management App is to save the time by making the process of order creation, products and invoice management easy for the user. It allows the users to create order with user friendly User Interface. After creating the order they are able to add products to that order, they can add the quantity and discount given on the products. After Order Creation users are allowed to create invoices based on the approved order, and notification would be sent to the associated people about the order status automatically. It also allows the manager of sales users to view the whole data in a simplified and attractive way.

Scope:

Inscope:

The functionality provided by the system is as follows:

This App includes 3 types of users which are business users, sales users and support team. Each of these would have some exclusive privileges.

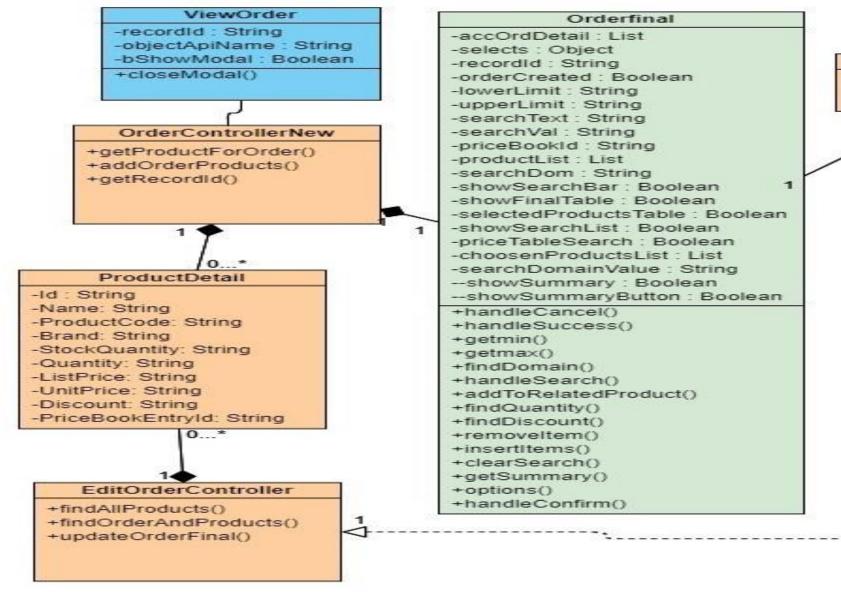
- 1. The Sales users can: (a). Create orders (b). Add products to each order with desired quantity and discount (c). Create Invoice of the approved products. (d) not see other sales users details (e). Manage up to 5 accounts (f). Receive email about order info.
- 2. The Business users can: (a). Handle up to two sales user. (b). Manage up to 10 accounts (c). Approve order if the order amount is greater than \$100000 (d). View the accounts and related Contacts, Orders and Invoices (e). Not edit or delete the data in the system. (f). Receive notifications of order status
- 3. The Support Team can: (a). Create business users and sales users (b). Assign the users to respective roles (c). See whole data under this app (d). Receive notifications of order status

Outscope:

The functionality which are not covered are as follows:

- The application does not cover shipment details.
- 2. It doesn't cover the method by which payment is received.
- 3. It doesn't give information about the delivery date of the order.
- 4. It doesn't have customer order request page
- 5. It doesn't send notification when the product is out of stock.

CLASS DIAGRAM



Order Summary

+getAccountDetails() +updateOrderDetails()

EditManager

-record I String
-newOrderSummary: String
-modeToggle: Boolean
-editModeToggle: Boolean
-tableToggle: Boolean
-addItemToggle: Boolean
-showSearchList: Boolean
-searchProductList: String
-priceBookId: String
-order: Object
-productList: List<String>
-toDeleteList: List<String>

+connectedCallback()

+editMode()

+addItemClick()

+closeModalTab()

+handleSearch()

+addNewProduct()

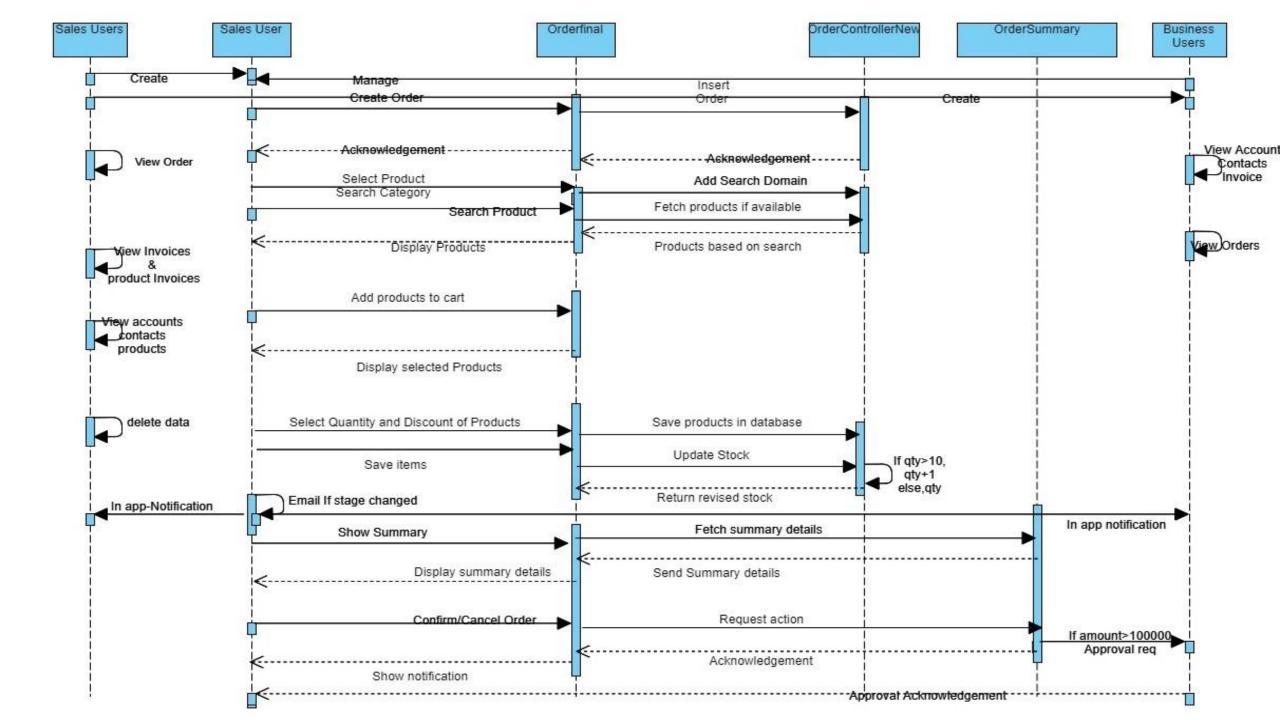
+deleteClicked()

+qunatityUpdate()

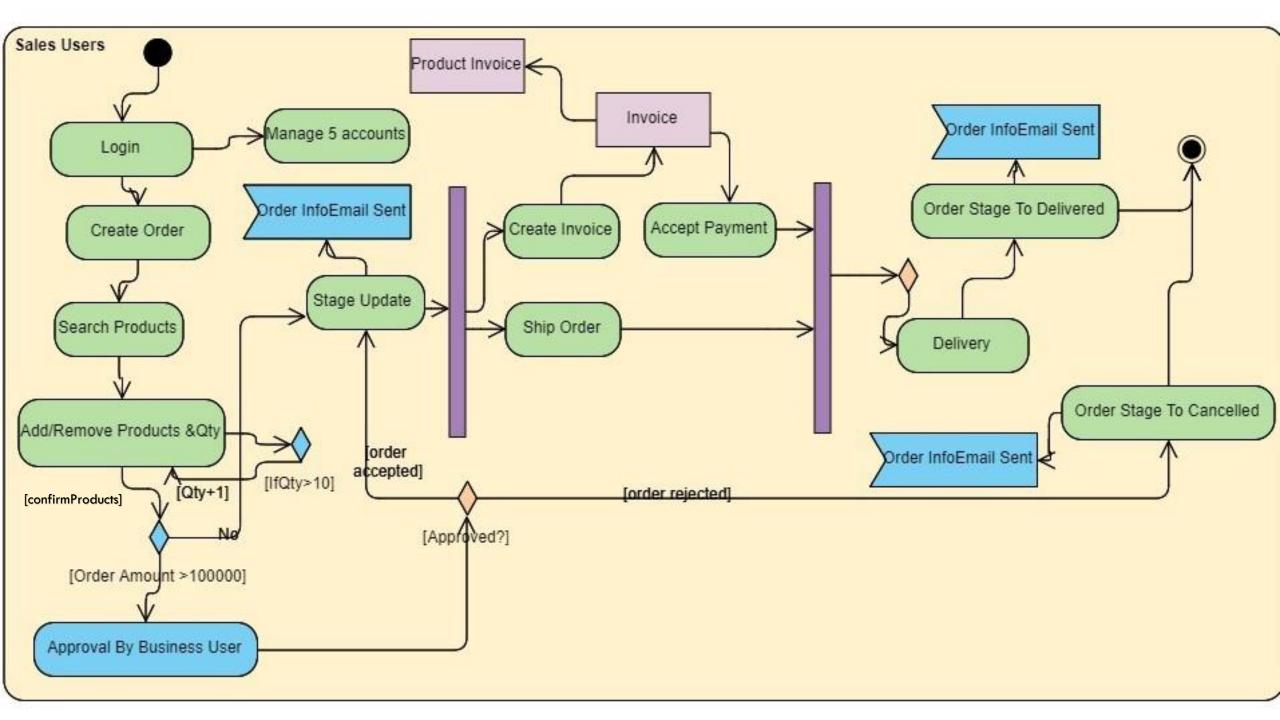
+discountUpdate() +saveClickedFinal()

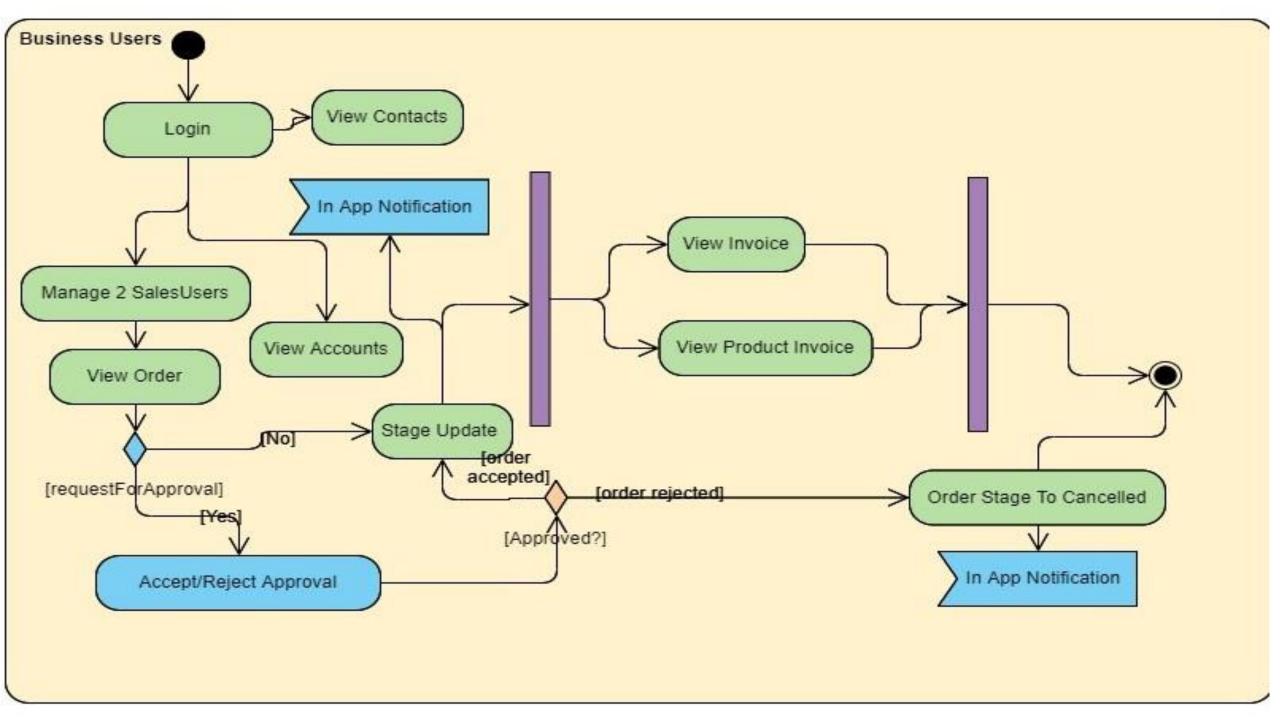
+cancelEditClicked()

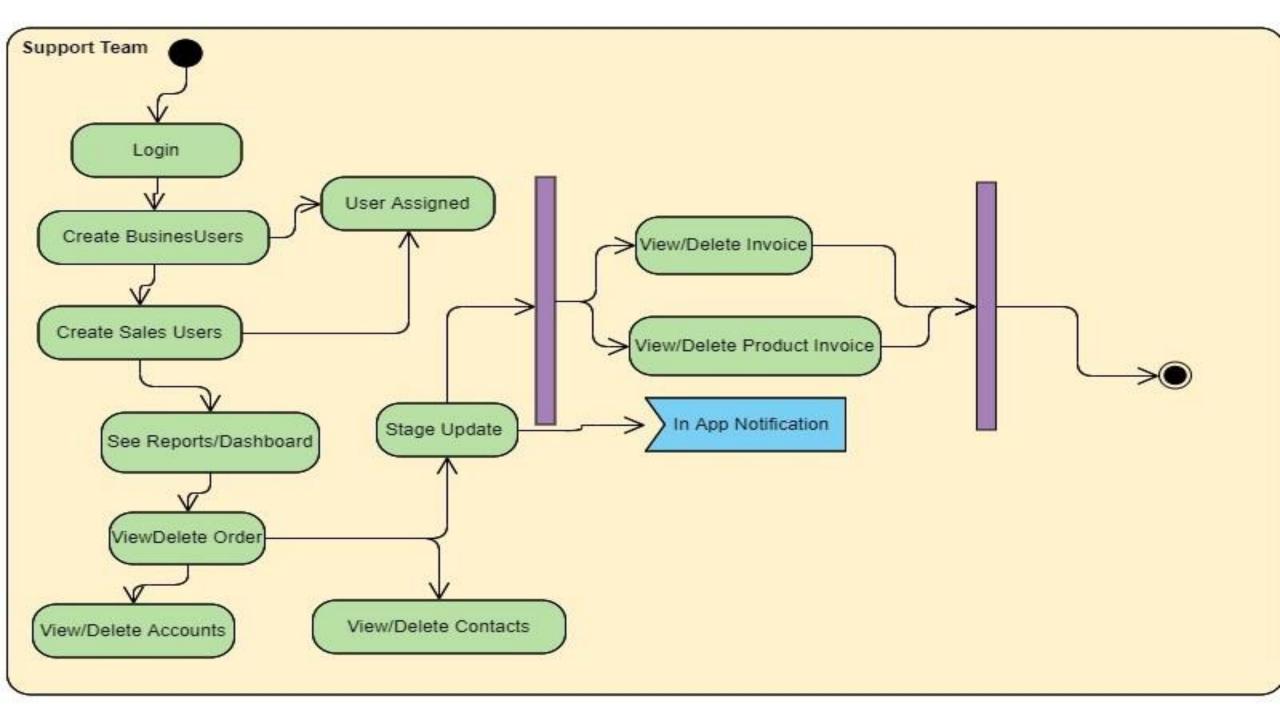
Sequence Diagram



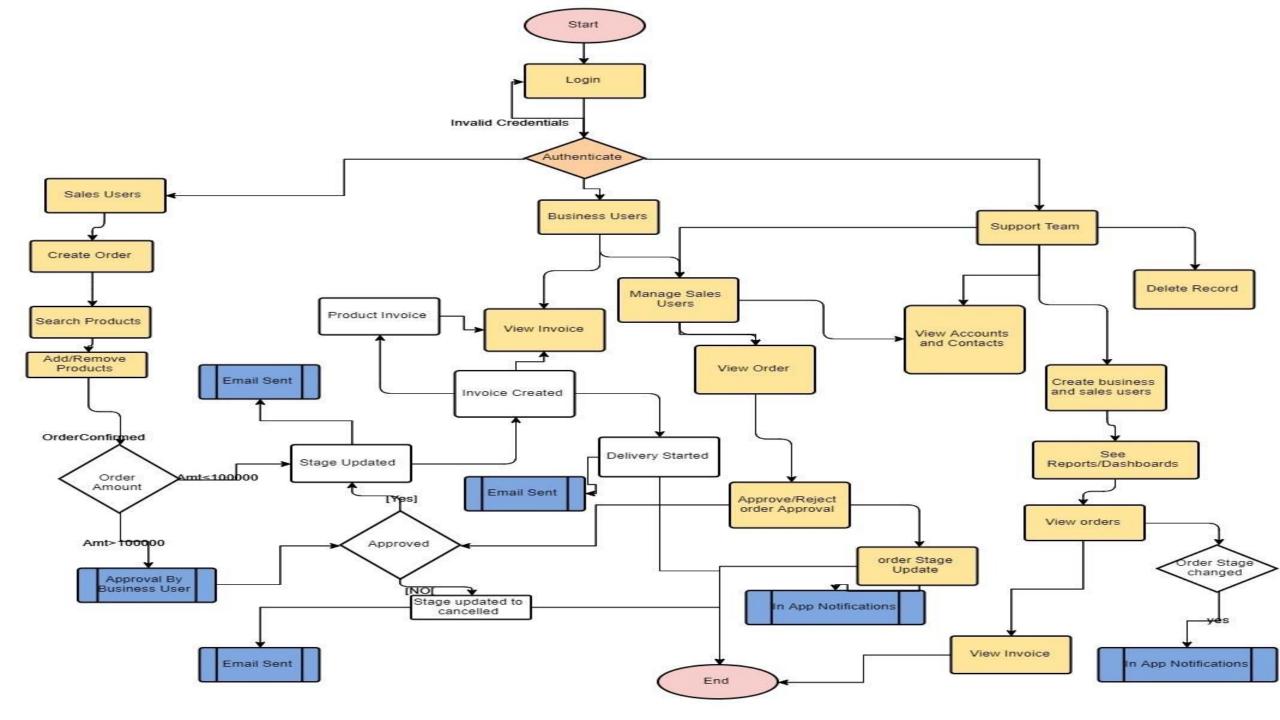
Activity Diagram



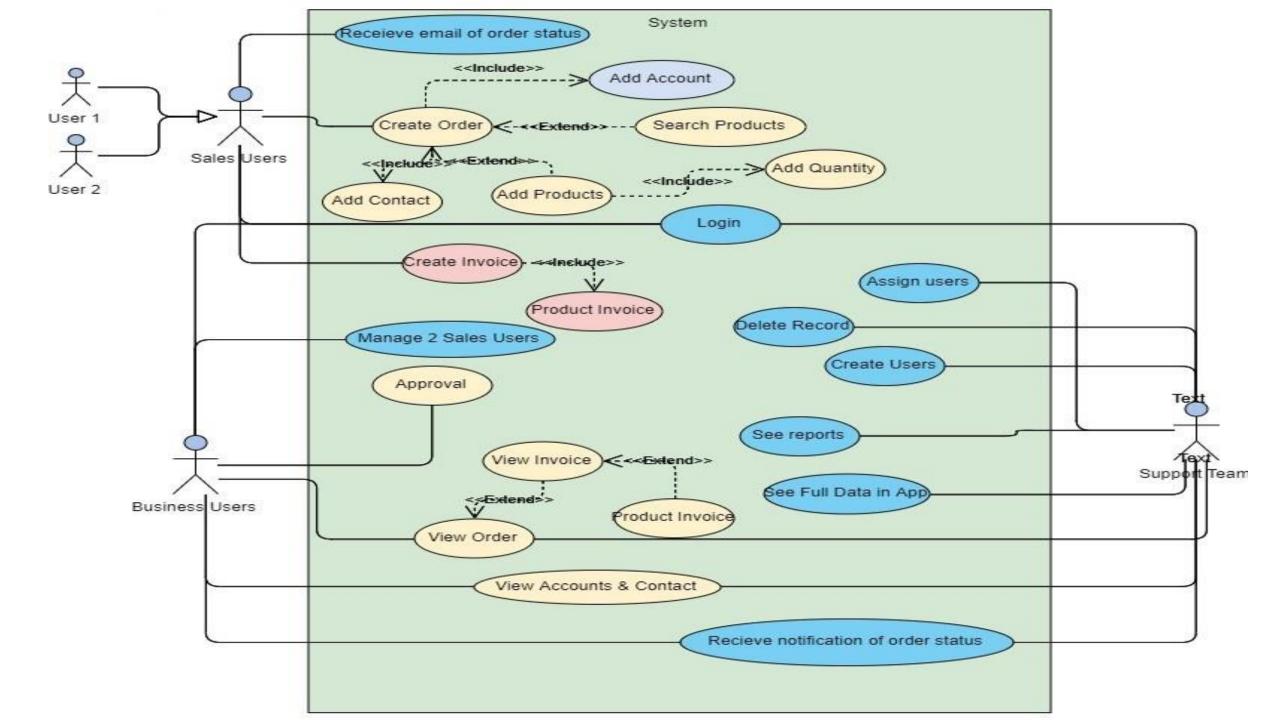




Flow Chart



Use Case Diagram



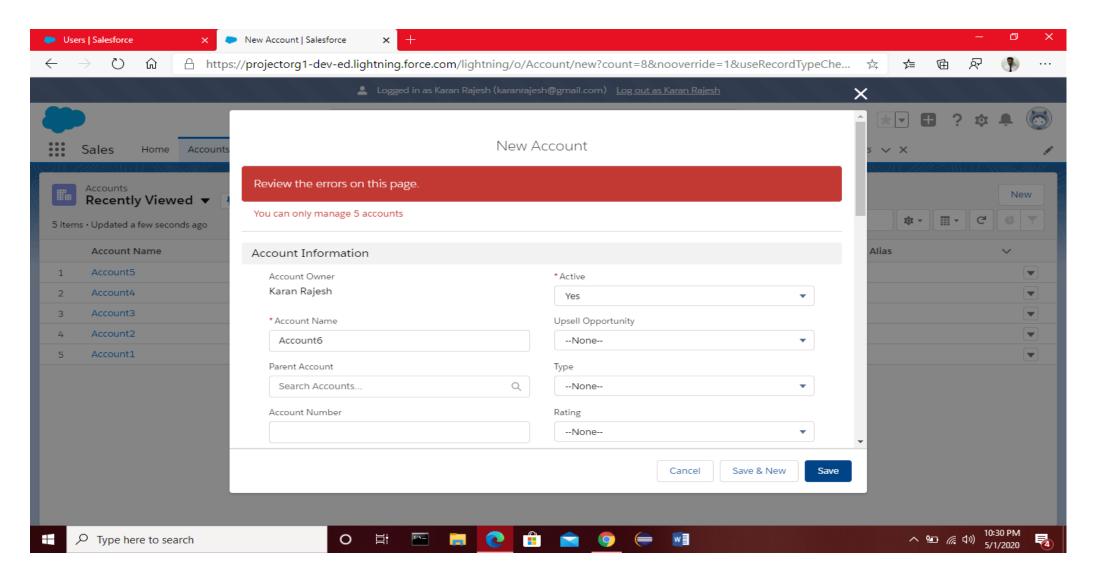
Wireframes

DEVELOPER PART

Triggers

1.To allow Sales User to manage at most 5 accounts.

Done with the help of trigger on account and using before insert and before update triggers. Getting all the accounts In a list and if the size of the list greater than 5 add error.



2.Email to those contact With email opt out field True

Getting list of accounts and contacts list of related account and sending message on stage change with the help of SingleEmailMessage

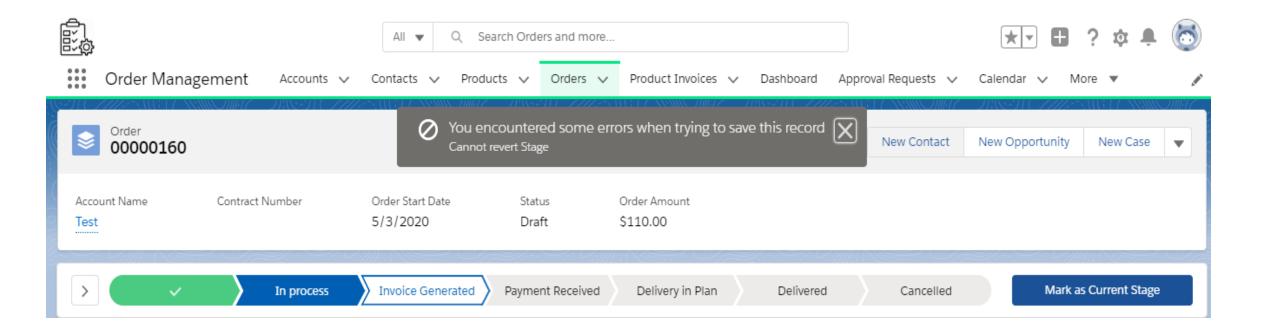


Order ID: 8012x00000003JEAAY Order Number: 00000147 Order stage: Created



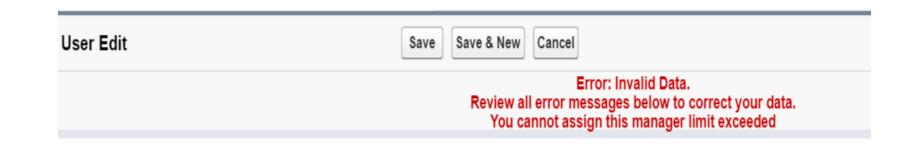
3.Stage can't go back to previous state

Trigger on order when there is change in stage.



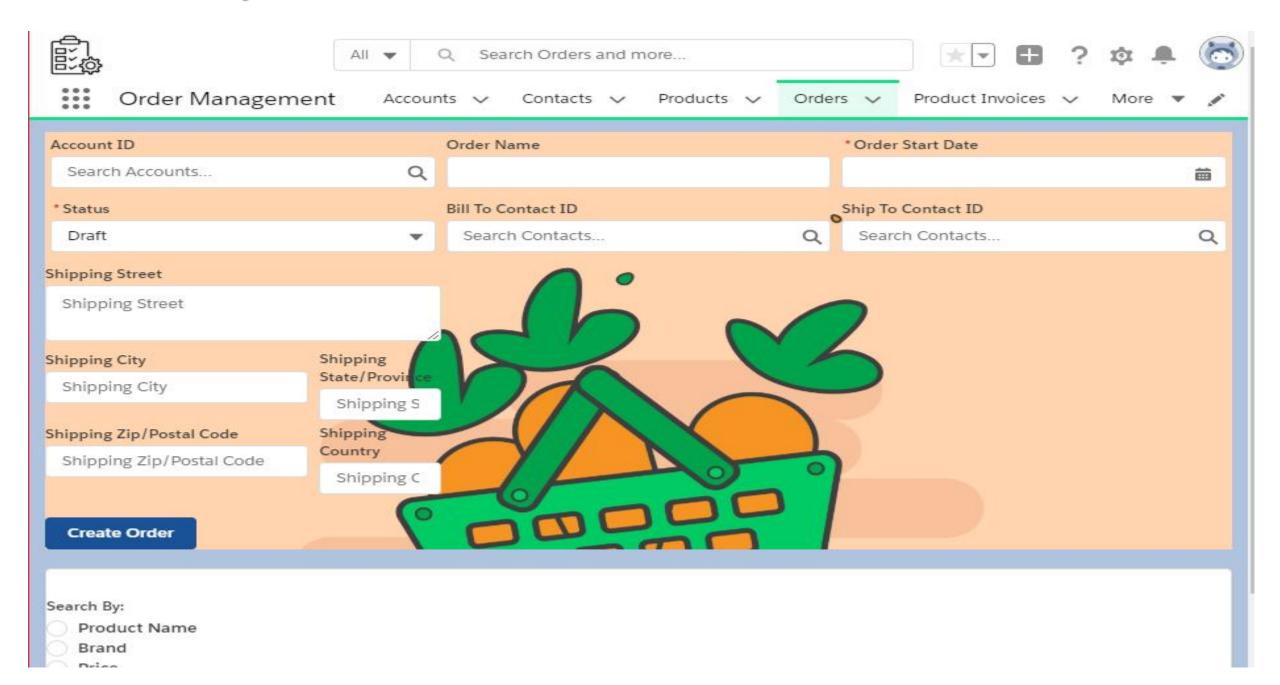
4.One Business User can manage at most 2 sales user

Trigger on User checking Manager Id Size.

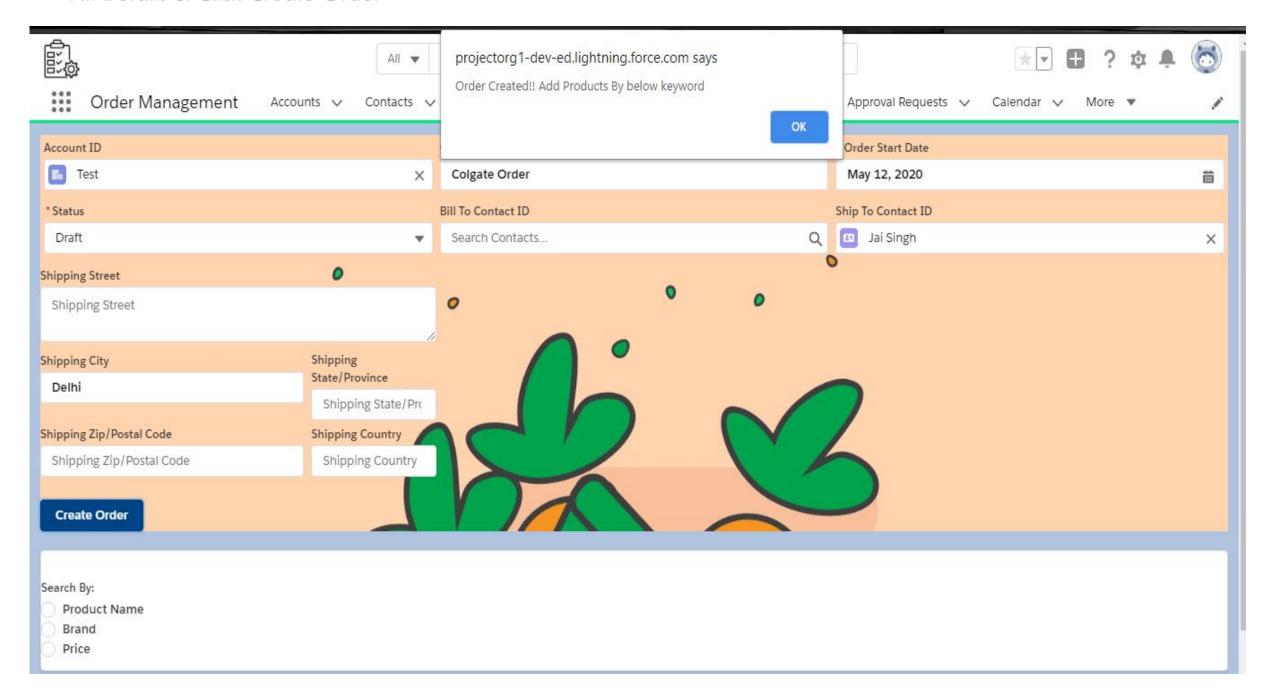


LWC & Aura

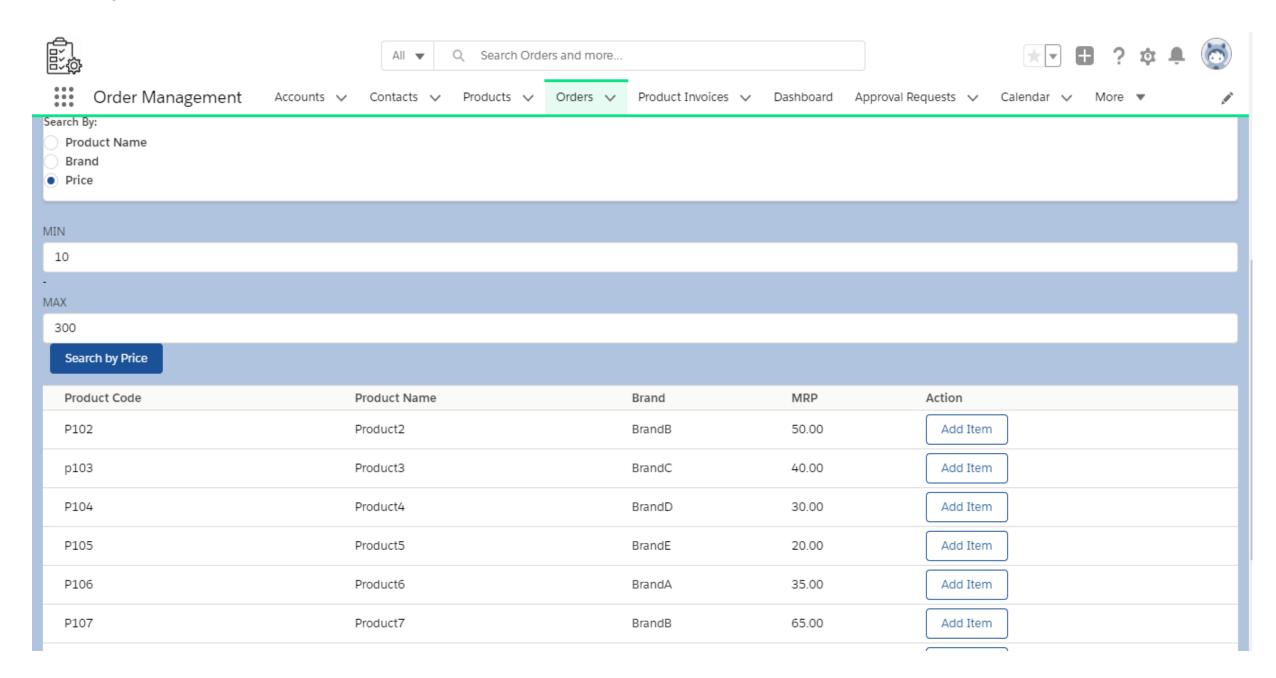
Order Creation Page



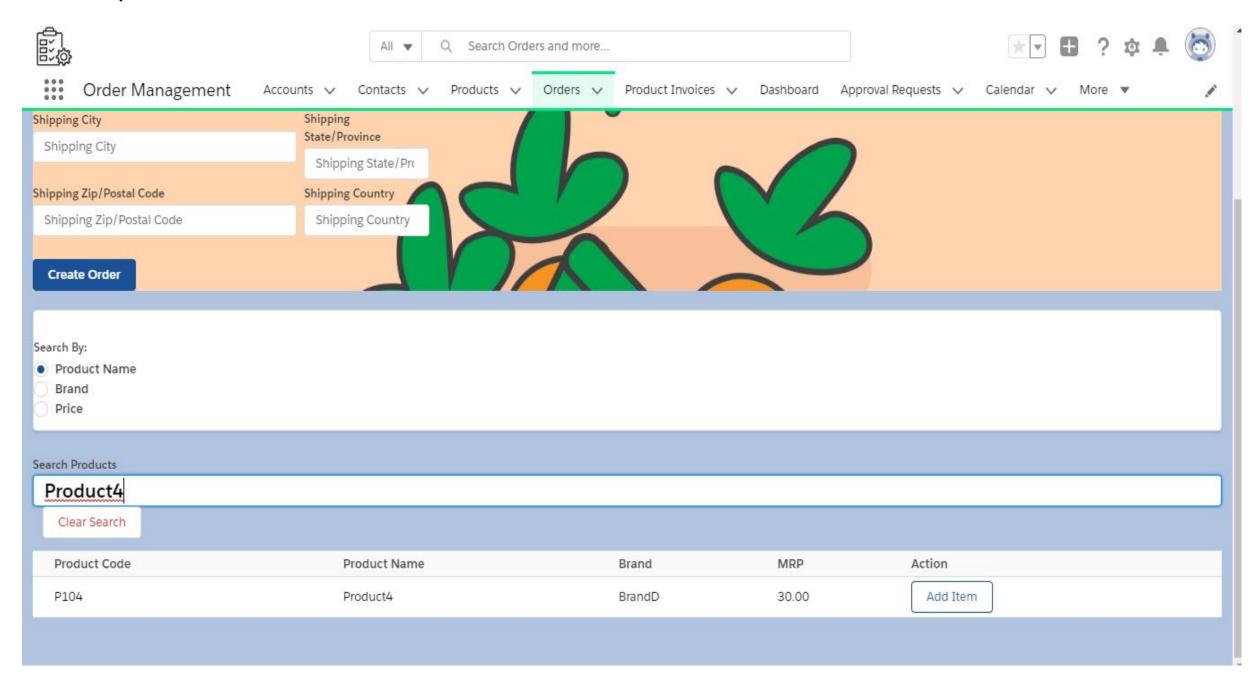
Fill Details & Click Create Order



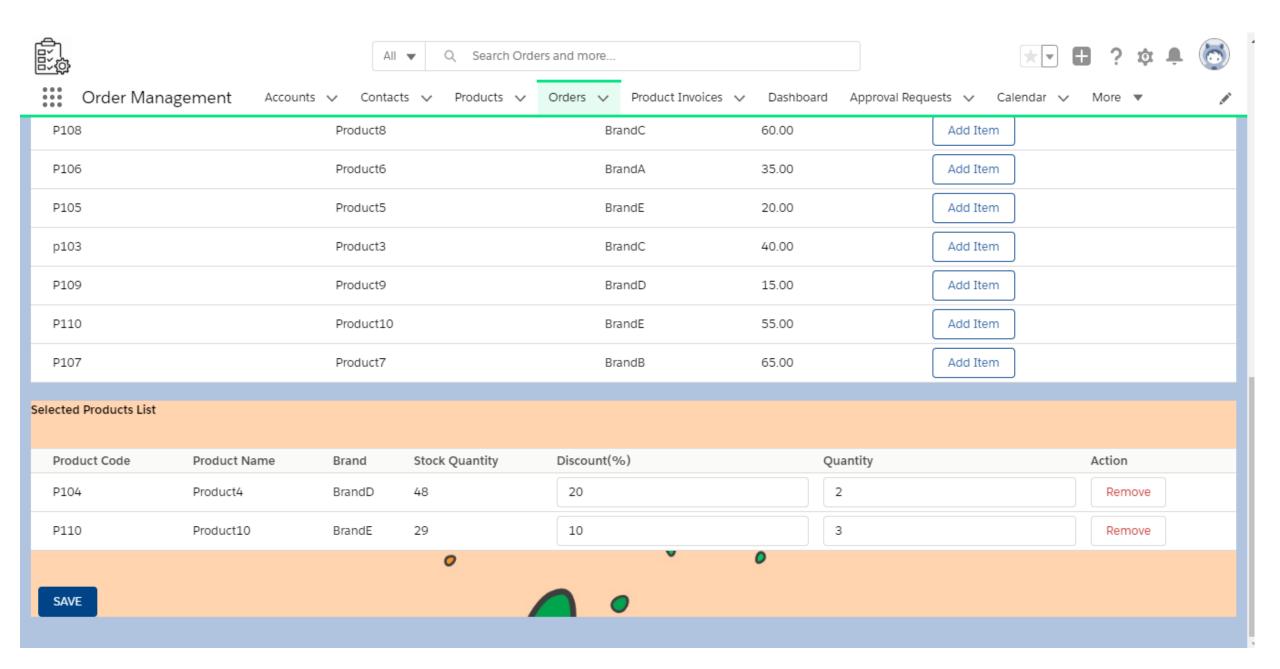
Search By Price



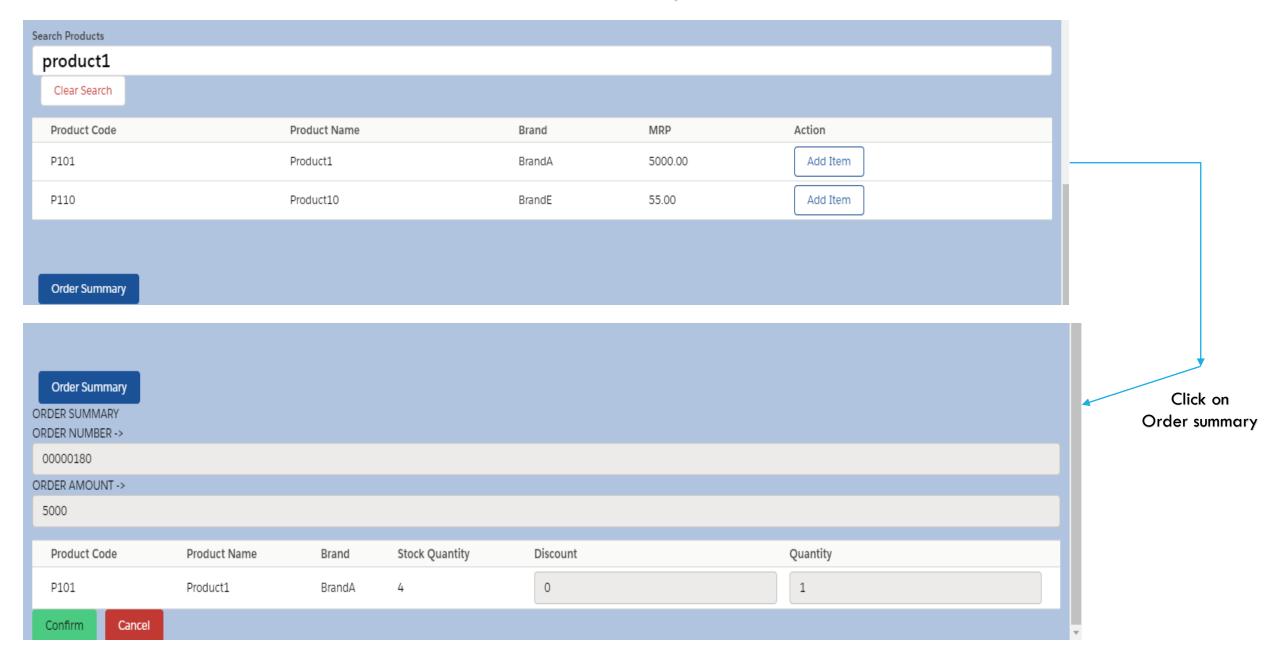
Search By Product Or Brand



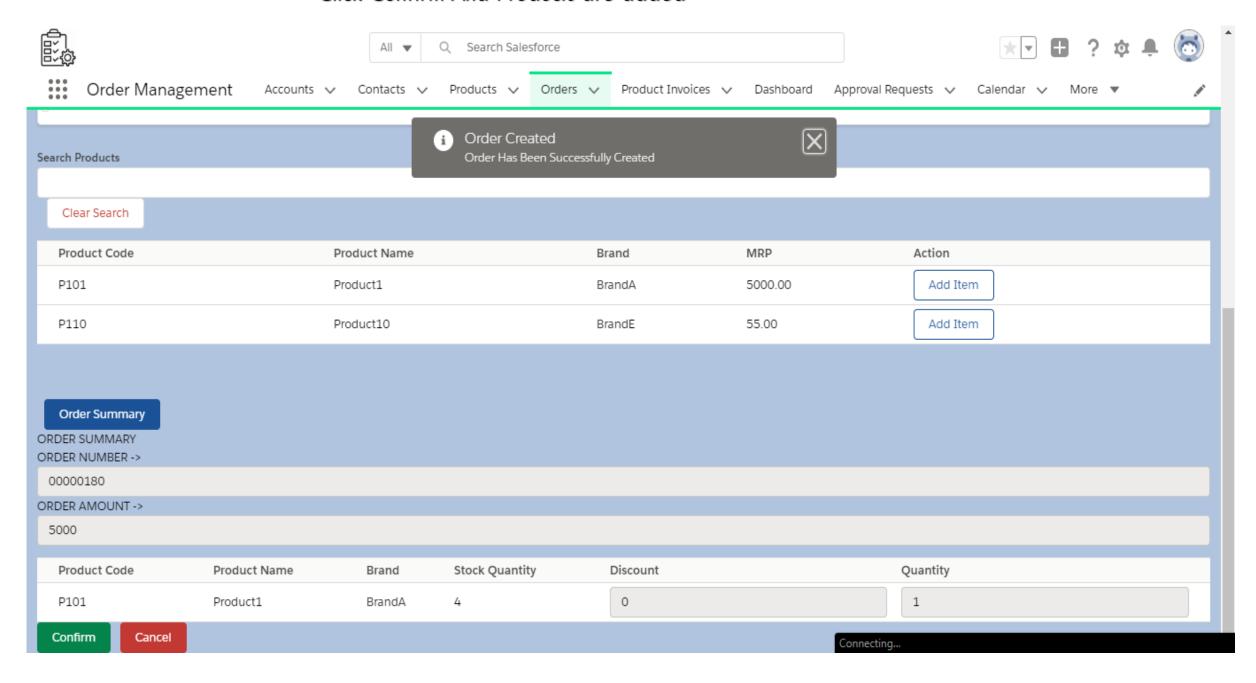
Select Items, then Add Quantity & Discounts, Click "SAVE"



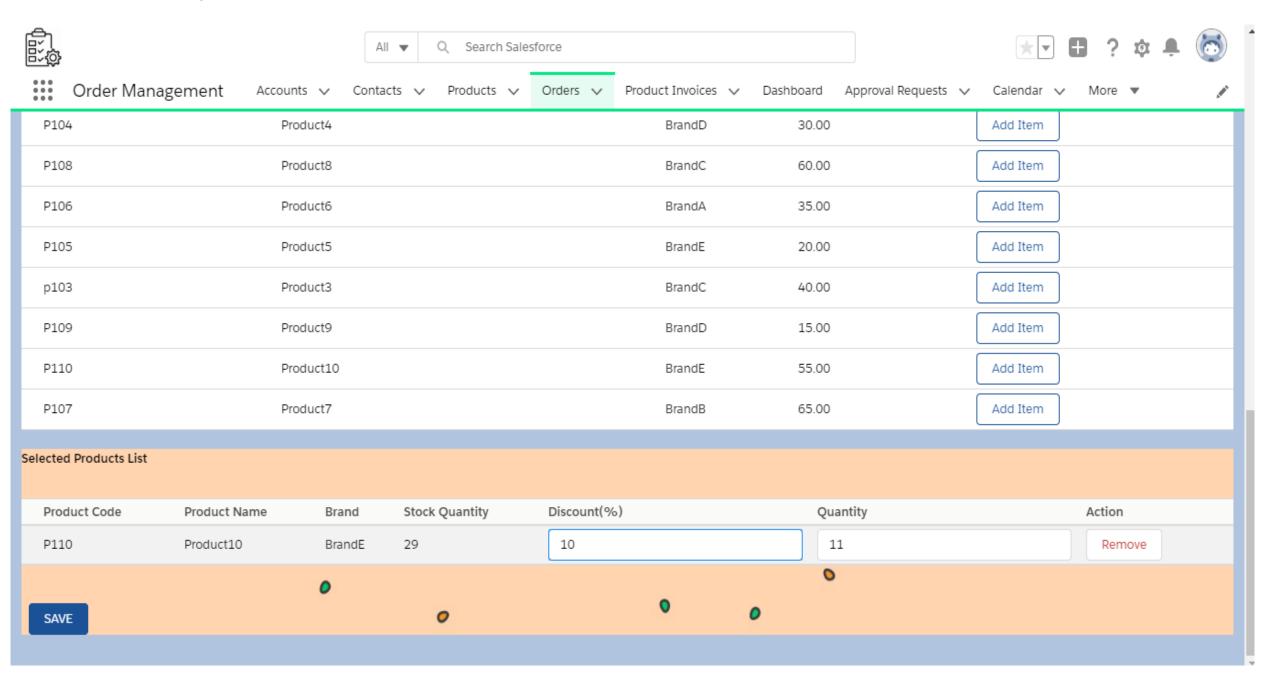
Click Save and then Order Summary



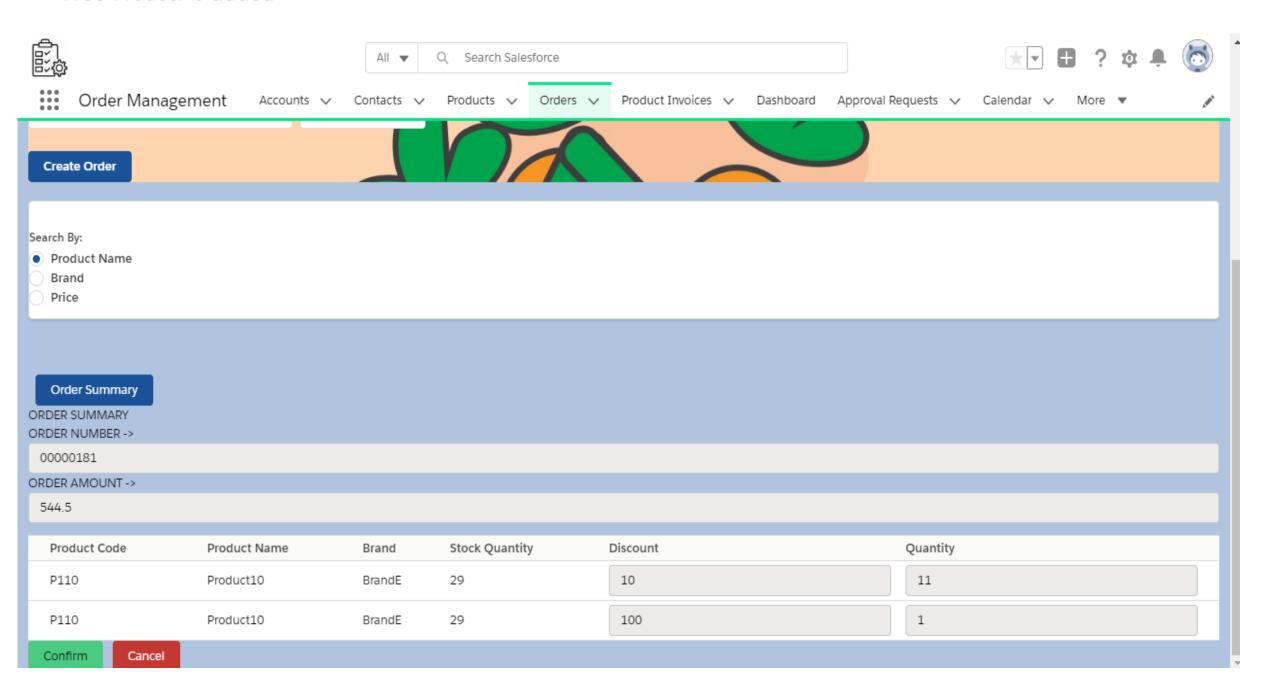
Click Confirm And Products are added



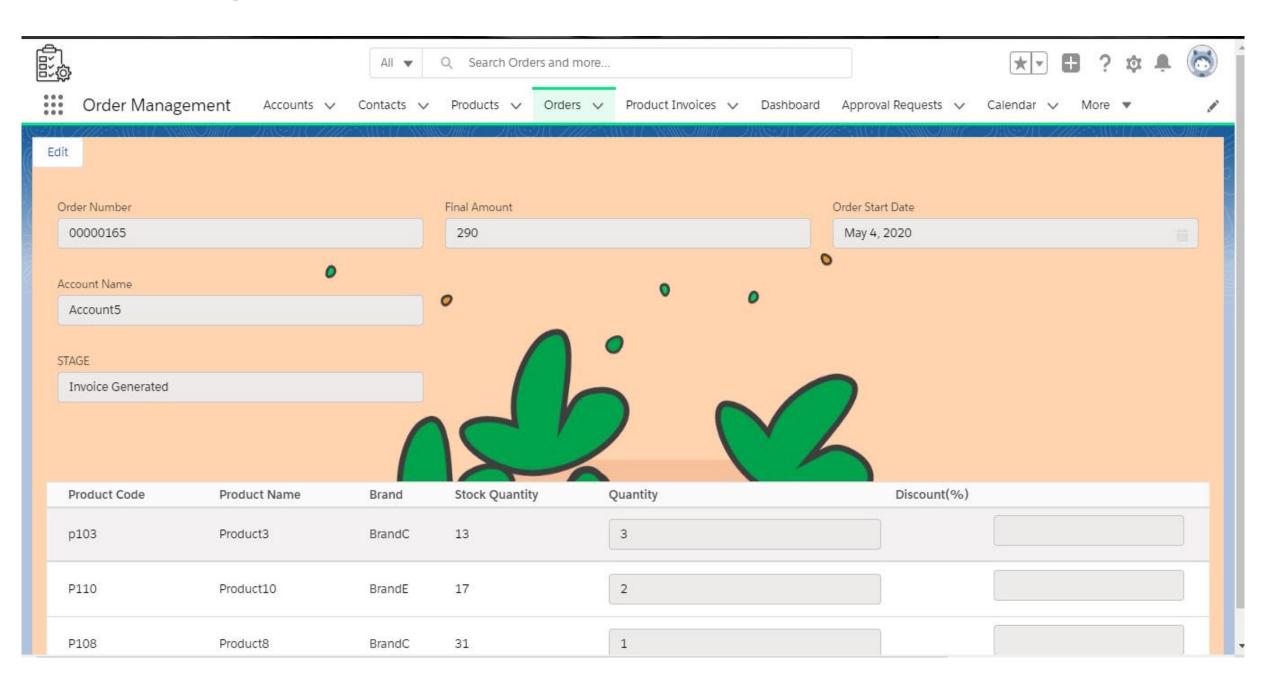
Add Quantity >10



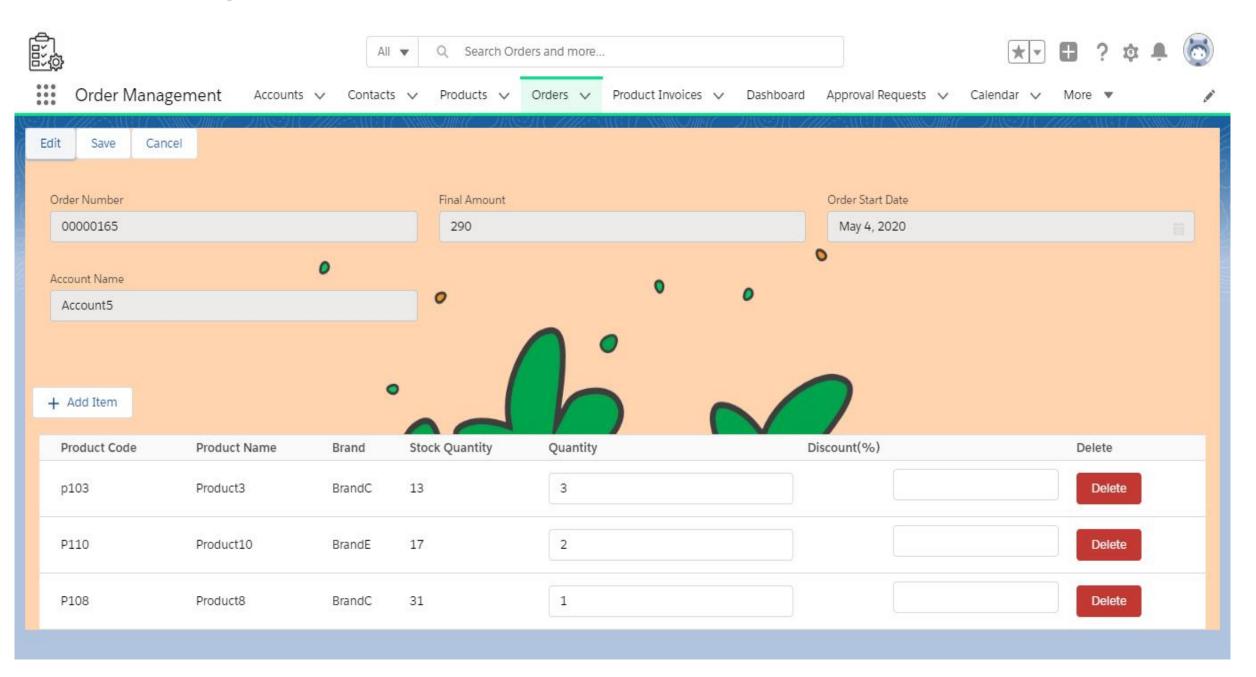
Free Product is added



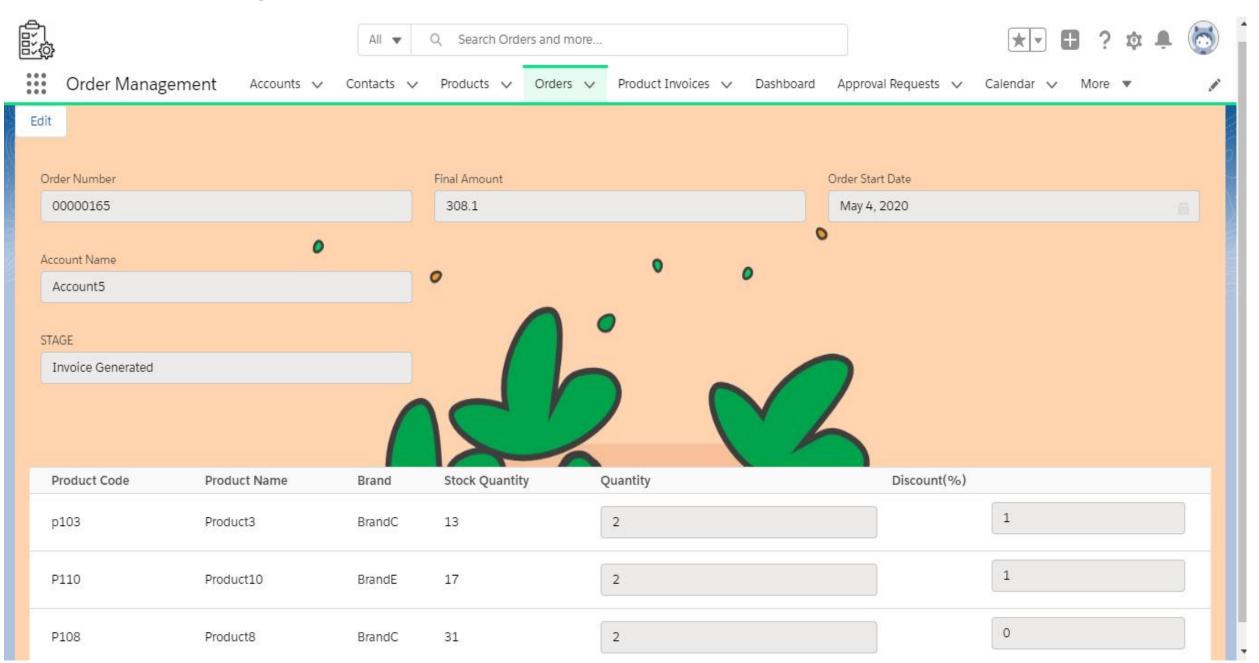
View Order Page



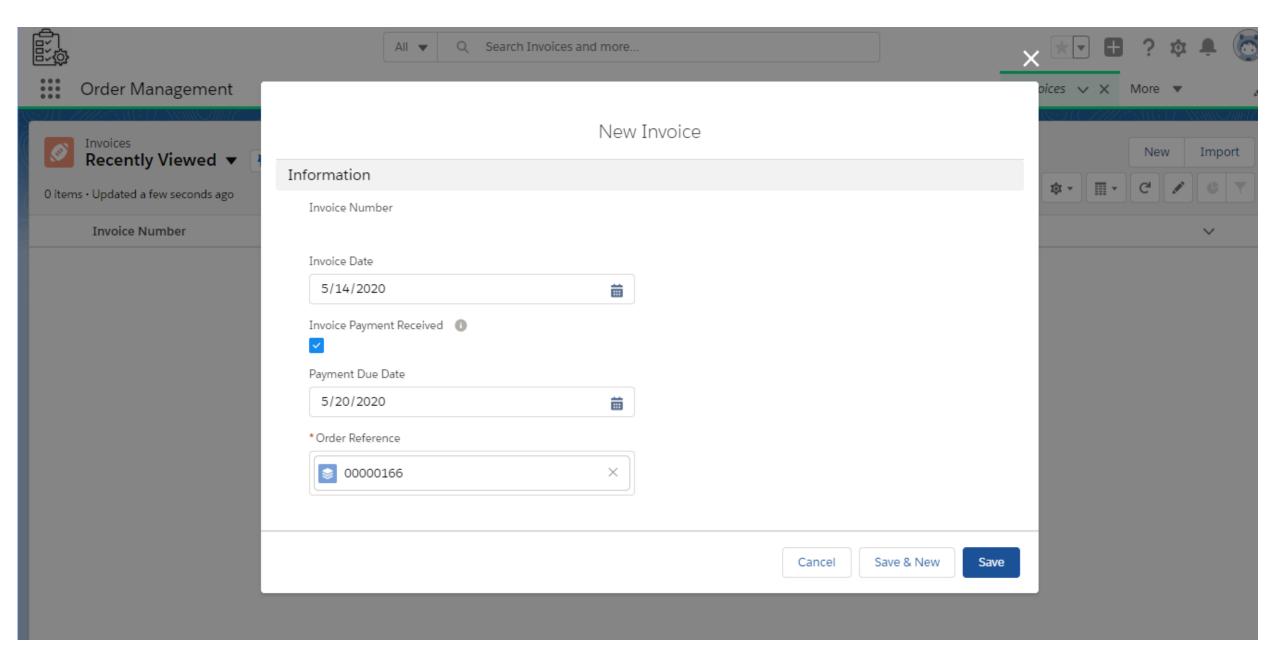
Edit Order Page



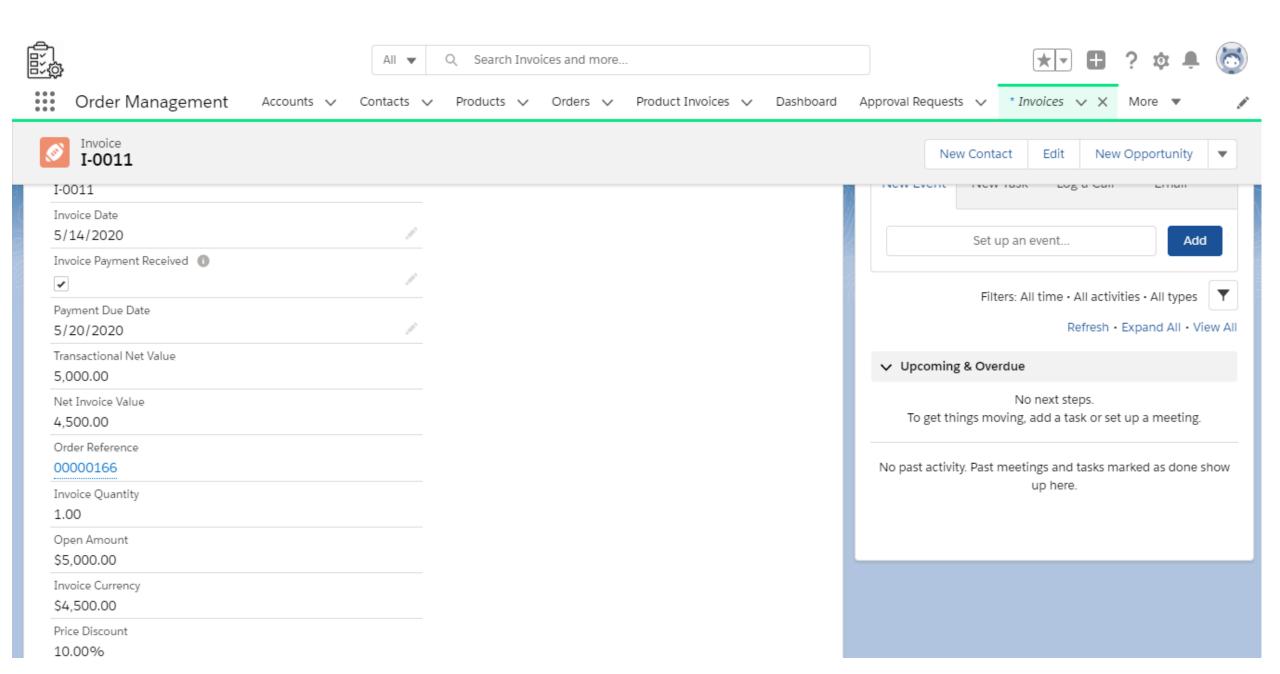
View Edited Order Page



Invoice Creation



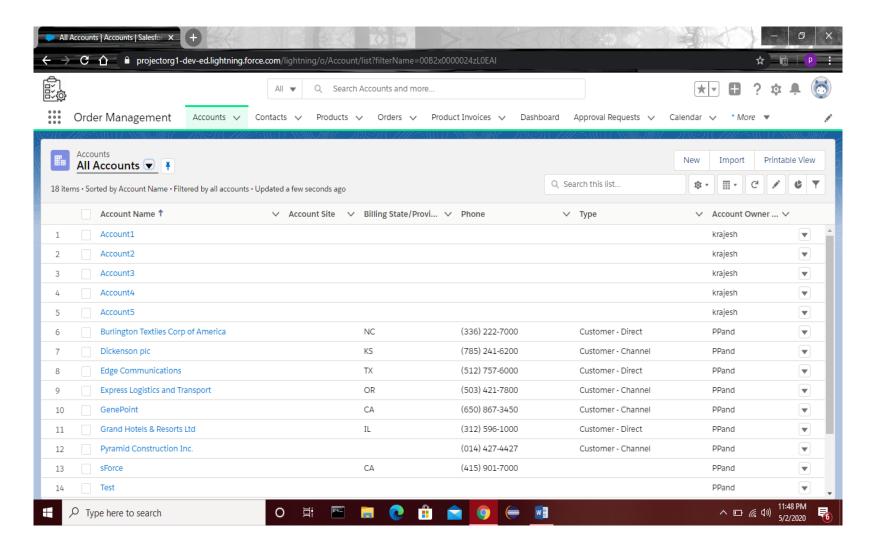
Invoice Auto Fields

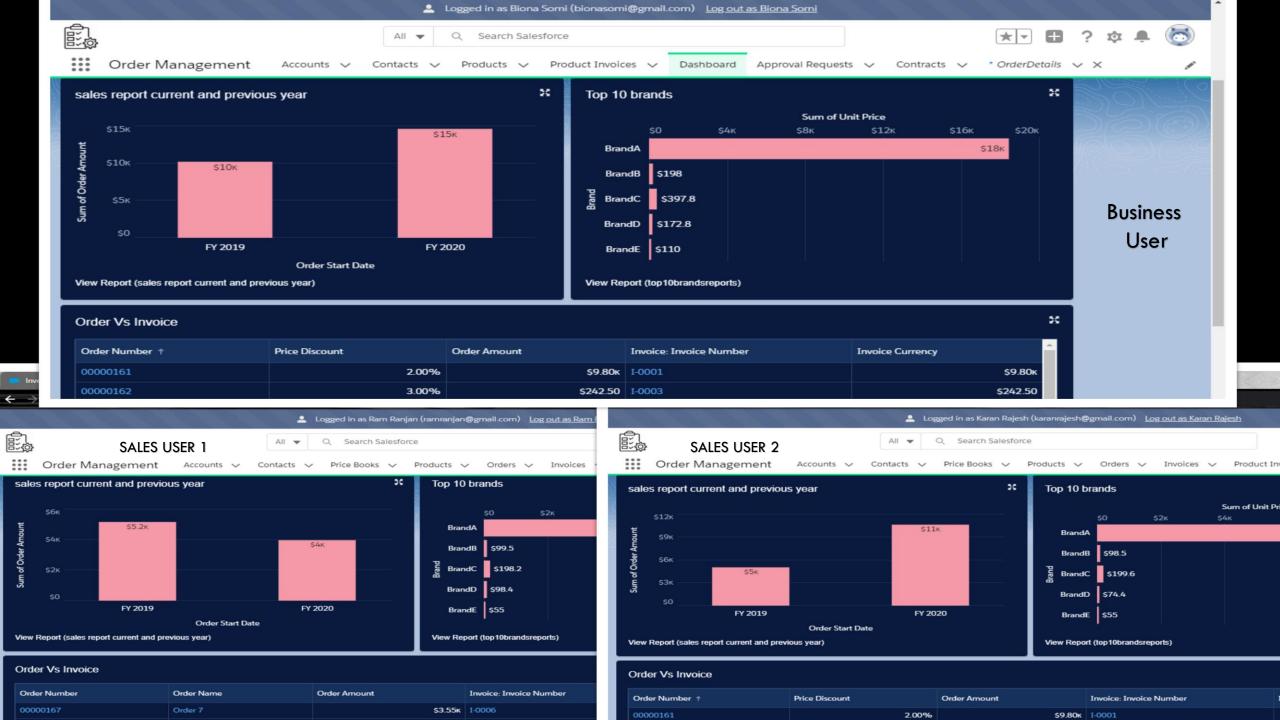


ADMIN PART

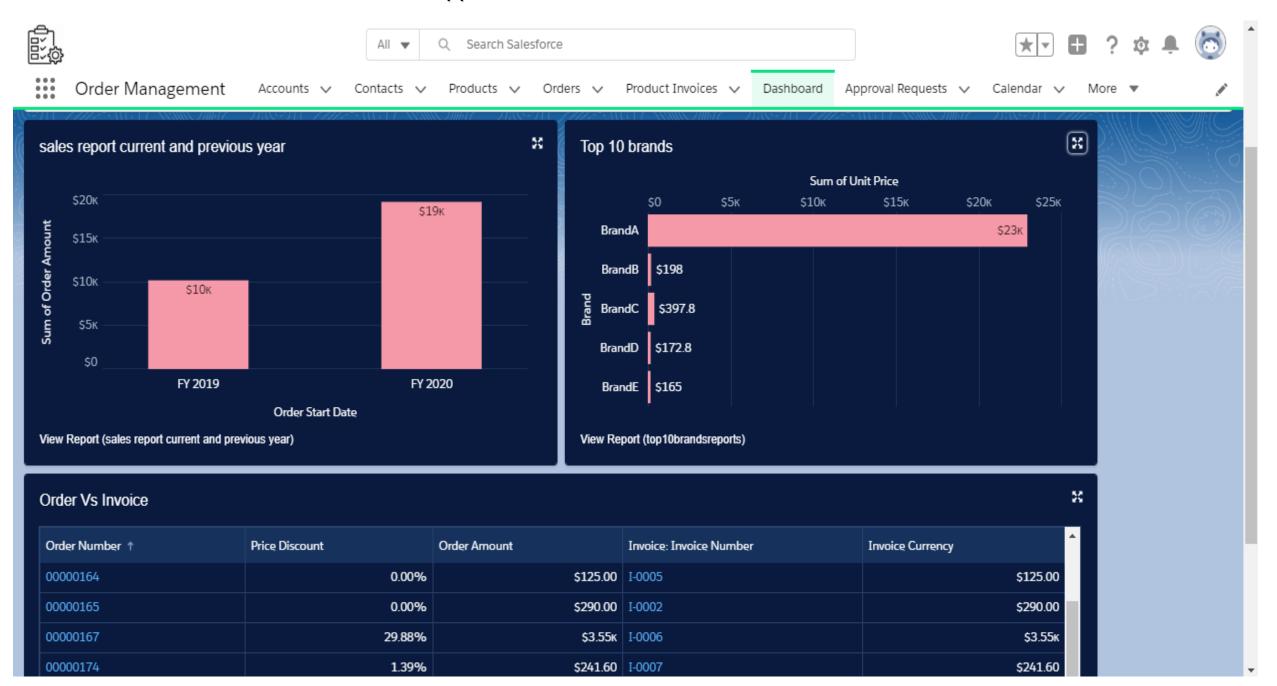
- 1. For allowing only support team to create users, uncheck manage users for business and sales profile.
- 2. Unchecking delete permissions for all the objects for sales and business profile to only allow support team to delete any record
- Going to sharing settings and changing org wide Default so that no two users should be able to see each others record. Making the required objects private.
- 4. Check Create order permission for sales user.

Order Management Lightning App





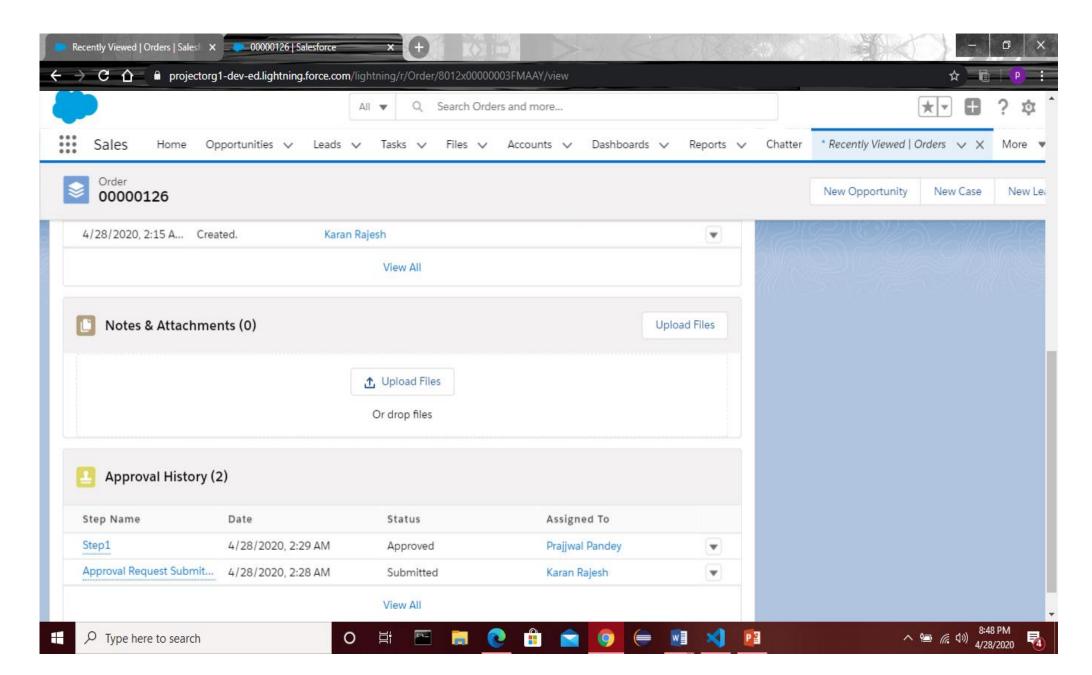
Dashboard For support team



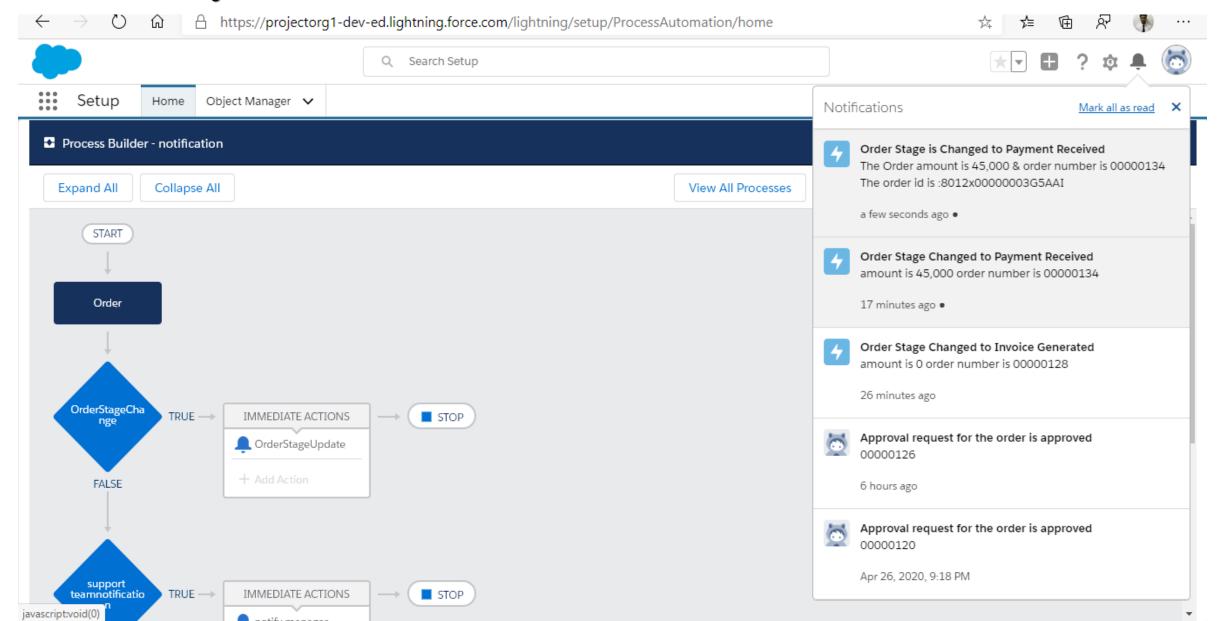
Process Builder

- 1. To send to approval by business user if amount is greater than \$10000.
- 2. To send email to the sales user on stage change
- 3. To update the stage if payment received is checked in invoice record and update stage to "Delivery in Plan" after one day of receiving payment.
- 4. Process Builder for sending in app notification to business users and support team about stage update.

Approval If order Amt>100000

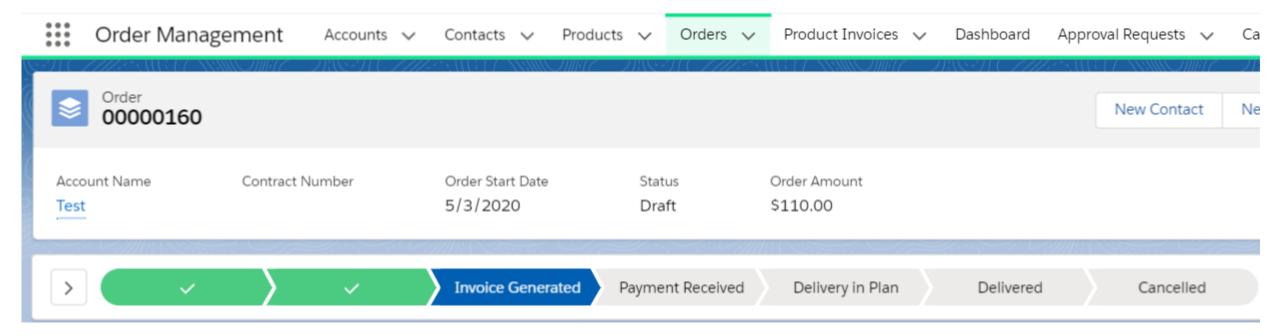


In app notification For Business and support team on stage change



Validation Rules

- 1. Order's End Date can't be less than order's start date
- Order's stage can't go back to the previous stage (INACTIVE)
- 3. Order remarks should be there if the stage is cancelled
- 4. Product stock quantity can not be less than 0
- 5. Invoice Start Date can't be less than order start date.
- 6. Payment Due Date in Invoice should not be less than Invoice Start Date



Class and Method Description:

- 1.OrderControllerNew: This class contains one inner class and methods to get the record id of order which is just created, to get product details and to add products to the order.
- <u>2.ProductDetail:</u> This class is inner class of ordercontrollernew and is used to store all the information of product selected by the sales user to add in the order.

Methods in OrderControllerNew:

- (a) getRecordId(): This method fetches the record id of the order created by the user.
- (b) **getProductForOrder()**:returns list of products in a JSON string format based on the search domain and search value. It gets the detail of products, pricebook, unit price and Id.
- (c) addOrderProducts(): This method gets the list of products selected by the sales user which is in Json string, it parses it converts it into product detail class and adds it into product detail list. Stage of the order is changed here from created to in process then the products are inserted with dml operation. It also updates the stock quantity of the product for both the cases i.e if quantity>10 or <10.
- **3.OrderSummary**: This class is used to display the summary page after adding the products. It will allow the sales user to confirm and cancel button.

Methods in OrderControllerNew:

- (a) **getAccountDetails()**: This method fetches the order number, total amount, account id to display on the summary page.
- (b) updateOrderDetails(): This method updates the stage if order is cancelled and confirms if the order is confirmed.

4.EditOrderController: This class contains one inner class product detail and methods to get fetch the products, get the order details and update the order details. The inner class product detail contains the variables to store the product details.

Methods in EditOrderController:

- (a)findAllProducts(): This method fetches product and pricebook information according to the search.
- (b)updateOrderFinal(): It gets list of products in a JSON string format based on the final edit and gets the detail of products which are removed by the sales user and converts into product detail list class. It performs two DMI operations, one to upsert edited items and one to delete removed items.
- (c)findOrderAndProductsd(): This method gets the list of all the products, total order amount and account name in the order which were displayed to the user.
- 5. OrderFinal: This is to used to create order, add products to the order, confirm and cancel order.

Methods in OrderFinal:

- a) handleSuccess(): This method fetches the record id of the order created by the user with getrecordId method in class orderControllerNew.
- b) getmin(): fetches the lower limit value entered by the user and stores it in lowerLimit attribute.
- c) getmax(): fetches the upper limit value entered by the user and stores it in upperLimit attribute.
- d) finddomain(): fetches the domain selected by the user for the search criteria if the price domain is selected the display window is shown to search by price.

- e).handleSearch(): It gets the search text entered by the user and find the product details by fetching the details from getProductForOrder function of class OrderControllerNew and stores it in the productList attribute in JSON format. This will make it show the list if there is data in productList.
- <u>f</u>).addToRelatedProduct(): Run a loop to enter all the data in the productList to the selects list. This will also display the data in selects in an area below it to display selected products.
- g) findQuantity(): To fetch the quantity for each product in a variable.
- <u>h</u>).findDiscount(): To fetch the discount for each product in a variable.
- i).removeltem(): If user clicks on remove item, it will splice the item from the list of choosenProductsList.
- **j).insertItems():** This will add the free product if quantity is greater than 10, and also calculates the unit price which is the price after discount and add the products to that order by using addOrderProducts method of class OrderControllerNew. It will also display the summary table.
- k).clearSearch(): to clear value in the search bar entered by the user.
- **I).displaySummary():** This will display the summary details of products selected by the user displaying the account name, order amount and product details.
- m)handleConfirm(): This will confirm the order and stage is changed according to the order amount n).handleCancel(): This will cancel the order, updates the stage to order cancelled by using the method updateOrderDetails in the OrderControllerNew.

6.EditManager: This is used to display the order details and edit the order details.

Methods in EditManager:

- (a)connectedCallback(): This method gets the list of all the products, total order amount and account name in the order which were displayed to the user by running getOrderAndProducts of EditOrderController and passing the current record id to the function in the lightning aura component. This will store all the details in productList.
- (b)editMode(): This will display the edit page if edit button is clicked.
- (c)addltemClick(): This will display the list of products to add if add Item is clicked.
- (d)closeModalTab(): To close the modal box in which all the products are displayed.
- (e)handleSearch(): To fetch all the products related to what user has entered in the search bar by using getProducts method of EditOrderController.
- (f)addNewProduct(): This will add the user selected products in the productList.
- (g)deleteClicked(): While editing user can delete the already existing products. This method will insert the already existing products in the toDeleteList.
- (h)quantityUpdate(): This method is to get the quantity entered by the user and insert the quantity in specific index of productList of the products.
- (i)discountUpdate(): In a similar way discount is added to the respected index of the productList.
- (j)saveClickedFinal(): If save is clicked, Unit price is updated with the discounted price and updateOrderItems of EditOrderController is called to delete and add the respected products along with updated quantity and discount. (k)cancelEditClicked(): This is used to cancel the edit.

Validation

- The Order Stage can't go back to previous stage
- 2. 1 Business user can be manager of two sales user and not more than that
- 3. 1 sales user can create at most 5 accounts
- 4. One must enter remarks if the order stage is cancelled.
- 5. User can not add product if stock quantity is 0.
- 6. Stock Quantity of products can't be less than 0
- 7. Order end date must not be less than order start date

Assumptions

- 1. Sales User have to manually create invoice after order is confirmed
- 2. Invoice number should be 4 digit format
- 3. Status is set to draft when order is created
- 4. Business user will be assigned by the support team
- 5. Date should be valid