Admin -

1. Add announcements for the user from time to time
2. When an order is received, check the inventory and manually confirm the pieces that can be delivered.
3. Have the authority to waive off the delivery charges as and when required for a particular order
4. Have the authority to cancel orders at will or when the user does not want to go ahead with the delivery due to non-availability of all the ordered items
5. Way to put black marks against certain users who are prone to creating a ruckus and black-list them if required.
6. Keep track of the sales
7. Keep track of the items sold
8. Keep track of the sales on a daily, monthly basis
9. Ability to add a new product category, school, product within a category (like add product XYZ to the list of products for school ABC), available colors for accessories
10. Ability to modify the description of a product, images related to an existing product, prices, delivery charges, delivery charge waive off amount
11. Ability to add any offers on purchase
12. Ability to run queries on the database to quickly check the data

User -

1. Save all the data in capital
2. Check that there is no redundant data by making a search on all the entered values
3. Mention the working hours i.e. the hours between which the user can expect a reply
4. Keep a marquee to allow admin to push through any announcement (like expected holidays, or when the orders cannot/will not be serviced)
5. When the user places an order, mention that the user will get a confirmation call if the entire order cannot be serviced due to some reasons.
6. Terms and Conditions to be robust
   1. Exchange rules
      1. Only on shop
      2. Preferably on a week-day
      3. Un-used
      4. Within a week of purchase
   2. Should not be used if you intend to exchange/return
   3. No bargaining with the delivery boy
   4. No return through the delivery boy
   5. No alterations on the pieces delivered (unless shop is visited)
   6. Order received can be cancelled or partly delivered in case of lack of inventory. User will be informed beforehand

Product Categories

1. School Dress
   1. Shirts, Pants, Skirts, Ties, Belts, Socks, lab coats, dividers
2. Accessories
   1. Swimming googles, head cap, Shoes, cycling shorts
3. Sports Items
   1. Swimming trunks, judo dress, cricket dress

Portal

1. Categorization based on the Items (dress, accessories, sports items)
2. When school dress is selected –
   1. Select the school
   2. List the available products
   3. User selects the product and then the required size
   4. Add it to a shopping cart
3. When accessory is selected
   1. Select the accessory and choose a color (?)
   2. Add it to the shopping cart
4. When sports items is selected
   1. Select the item to be bought and the required size
   2. Add it to the shopping cart
5. Once all the products have been selected, ask user if he/she wants to buy any other product. If yes, redirect the user to category selection page. If no, send the user to the check-out page
6. On the check-out page, ask the user to review the selected products
7. Ask the user to log-in, if not done already or sign-up. Direct the user accordingly
8. Once done, ask the user to review and confirm the delivery address. Ask the user whether he/she wants to change the delivery address
   1. If yes, ask the user whether they want to update their profile with the new address provided
9. Once address is confirmed, show a success page saying the order has been received.

How it can be done

1. Separate folders to keep the images and other specific content for a school and a product therein
2. For searching the image, the path to be stored in DB table with columns as Product Category Code, School, Product, Location (will help if there are some common content across schools like plain white shirts)

**DB Design**

|  |  |
| --- | --- |
| TABLE | COLUMNS |
| Users | UserId, Firstname, lastName, gender, loginId, emailId, dob, date\_added, date\_last\_update |
| Login | Loginid, password |
| Address | addressId, streetName, city, state, pincode, Guest(Y/N), |
| Items | itemId, itemName, price, imagePath, productCategoryId, subProductCategoryId |
| productCategories | productCategoryId, categoryName |
| Orders | orderId, orderReceivedDate, orderStatus, cancellationReason |
| OrderDetails | orderId, itemId, addressId |
| subProductCategories | subProductCategoryId, subProductCategoryName, productCategoryId |

OrderStatus – Order Placed (1), Order Dispatched (2), Order Delivered (3), Cancelled by user (4), Cancelled by seller (5)

**Suggested Change**

|  |  |
| --- | --- |
| TABLE | CHANGE SUGGESTED |
|  |  |

**Tech Stack**

ORM - Hibernate

Framework – spring mvc

All Rest calls

Front end – angular JS

Database – independent (mysql, oracle)

**SUGGESTED CHANGES**