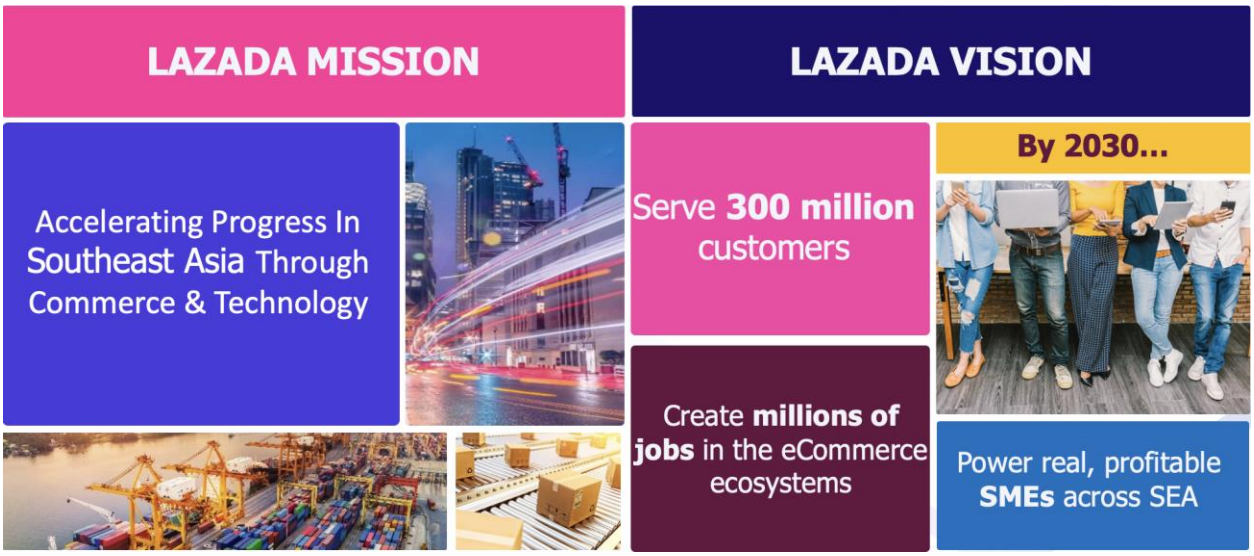
Lazada

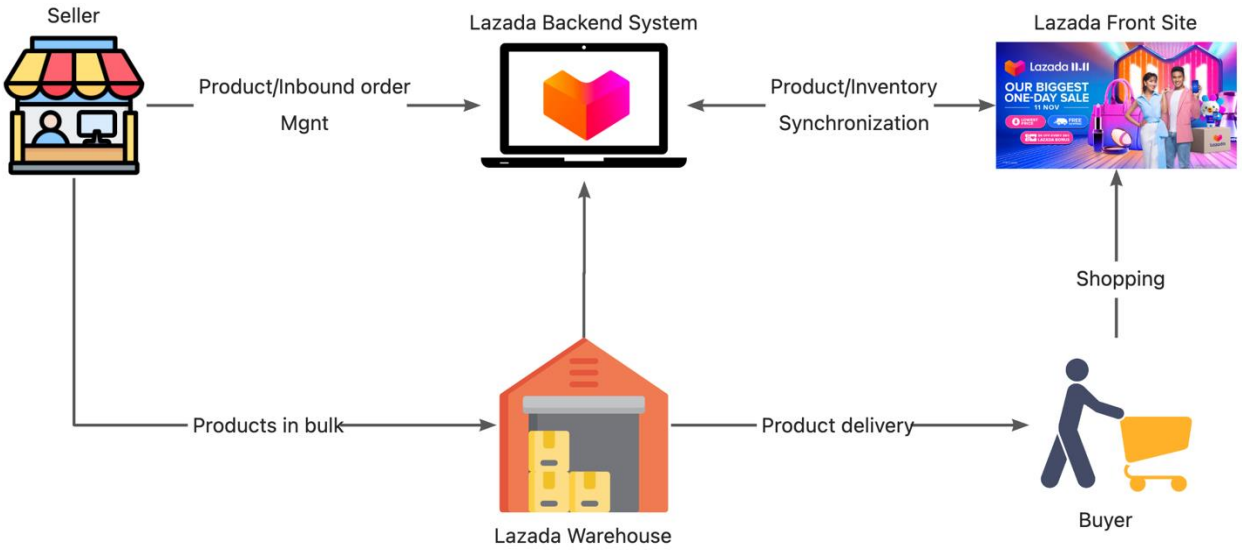
Background

Founded in 2012, Lazada Group is Southeast Asia's leading eCommerce platform. With a presence in six countries – Indonesia, Malaysia, the Philippines, Singapore, Thailand and Vietnam – we connect this vast and diverse region through our technology, logistics and payments capabilities. Today, we have the largest selection of brands and sellers, and by 2030, we aim to serve 300 million customers. In 2016, Lazada became the regional flagship of the Alibaba Group, and is backed by Alibaba's best-in-class technology infrastructure.



ECommerce System

Besides providing marketplace for sellers, Lazada also provides drop shipping business model where seller's products are stored, picked, packed and shipped directly from Lazada warehouse to ensure your customers enjoy the fastest, most convenient delivery, it's FBL – Fulfillment By Lazada.



You can also checkout detail about FBL here

General Requirements

In this use case, we will design a minimal simplified FBL model

We will look at five roles in this model:

- Customer: who buy products from Lazada
- WH Administrator: who manage product category and warehouse (WH) area setup
- Seller: who manage product, inbound order (IO) creation
- WH Worker: who work in WH and take care of order fulfillment
- Shipper: who ship products from warehouse to customers

To be able to operate this FBL, we need product solutions for each entity involve in this model

Lazada Site:

Customer: customer a need a website where they can find products they need and make purchase

- Browsing, filtering products by product's attribute for example price, brand, color and extra attributes from product's category
- Search product
- Do checkout
- Make order

Admin System:

WH Admin: WH admin needs product solution to manage product category, warehouse areas

- Product Category management: create, update, delete. Team can define additional product attributes at category level then each product belongs to product category will have those extra attributes
- Warehouse area management: create, update, delete. Each area has CBM info. One product can be stored in many areas, total product's quantity is total of stock of product from all areas that product is in

Seller: Seller need product solution to manage product and IO creation

- Product management: create, update, edit, delete. Each product has basic attributes like: title, description, price, image and product dimension information: length, width, height. Beside product basic attributes, product also inherits attributes from product's category.
- Inbound order: seller can create inbound order for their products. Product inventory will only be updated when WH worker finish IO

WH Worker:

- IO listing: can view and finish IO, system need to recommendation areas to put for each product in IO and action to finish PO. Recommendation logic should prioritize for areas:
  1. Already have same product in
  2. Still have enough space to put whole amount of product in (base area free CBM vs product's CBM)
  3. Empty areas
- Order listing: can view, and pick products for order, system needs to recommend areas to pick for reach product in order. Recommendation logic should prioritize areas that have enough quantity for product in an order

Delivery hero:

- Order listing: can see list of ready-to-deliver orders and delivery order action

Specific Requirements for ISYS2099 – Database Applications

The following requirements must be addressed in the group project. Your application only needs to support three roles: Warehouse admins, sellers, and customers.

Database Security

Create appropriate DB user accounts and roles with the necessary privileges to interact with the DB server. Explain the reasons behind your choice.

Warehouse Admins

Product Category management: Create, Read, Update, Delete. Updating and deleting a category is available only if no product is associated with that category. Each category has a name. A category may belong to another parent category or NULL if it is a top-level category. Warehouse admins can define additional product attributes for a specific category. When defining an additional attribute, the administrator can specify its name, whether required or optional, whether a number or text. A product belonging to category X has all basic attributes, all additional attributes defined in X, and all additional attributes defined in X's ancestor categories. Use NoSQL for this requirement.

Inventory management: There are multiple physical warehouses. Must support Create, Read, Update, and Delete warehouses. Each warehouse has a name, address (province, city, district, street, number), and total ~~area~~ volume (a number). Deleting a warehouse is available only if there is no product in that warehouse. Products can be moved from one warehouse to another. Use transactions to ensure the total number of products in all warehouses is consistent before and after a move.

Sellers

Product management: follow the requirements specified in the General Requirements section.

Inbound order: After sellers select a product and enter a quantity number, the system will automatically update the inventory of that product (i.e., there is no WH worker here) and decide the warehouse to stock those product items. To simplify, let's assume each product item has a total ~~area~~ volume of (width * length * height). Then, the system will choose the warehouse with the largest space available to stock those product items. If there is not enough space, the remaining product items will be stocked in the next warehouse, and so on, until all product items have been stocked or no more space is available in all warehouses. Use stored procedures to implement the warehouse selection logic.

Customers

Browsing products using product categories. Filtering products using price. Searching products with custom keywords (the search must look up information in the product title and product description fields). For product browsing, filtering, and searching, customers must be able to sort the displayed products based on price and added time (ASC and DESC). The performance of product browsing, filtering, and searching must be optimized using appropriate optimization techniques.

Add to Cart and Place orders (must check the product inventory to see if there are enough products to sell). Must use transaction & concurrency techniques to ensure the validity of orders (for example, if there is only one product item left, but two customers place orders for that product at the same time, only one can be successful).

Simulate the order delivery on the customer side. In the customer interface, for each placed order, there are two buttons: Accept and Reject. If this is an Accept, update the inventory information. If it is a Reject, send back the products placed in the order to the appropriate warehouses. Use triggers (combined with stored procedures) to automate the inventory update once an order status is updated to Accept or Reject.