

User Auth (Fire base)

PK: Primary Key (Unique)

- uid (auto gen)
- identifier
- created
- sign in (password will be encrypted here)

don't hv
to create
any of
these fields

User

PK

- uid (unique)
- type (consumer/wholesaler)
- + create User

Consumer

- uid (unique) PK
- first Name
- last Name
- email (unique)
- phone no. (unique)
- addr [similar to wholesaler address]

Note: I think
shd create
another table

ShoppingCart

- uid } PK
- uen }
- pid }
- qty

User Payment

- uid } PK
- card no. }
- expiry date }
- CVV }
- name on card }

Wholesaler

- uid (unique) PK
- company name
- uen (unique)
- email
- preferred currency (SGD/MRP)

Wholesaler Address

- uen PK
- street name
- unit no.
- building name
- City, country
- Postal Code

Order History

- order id (PK)
- uid
- uen
- pid
- qty
- price (per item)
- status (pending/completed/cancelled)
- rating (optional)

Product Supply

- uen } PK
- pid } PK
- price
- min order qty
- stock
- rating (either under
true/wholesaler)

Products

- pid (unique, auto gen) PK
- name
- package size
- img

Wholesaler Account

- uen PK
- full name (bank acc)
- acc no.
- bank.

Map API data

wholesaler + user
consumer (not product)

Currency API data

wholesaler uses

[In Pascal Case] - Func Names

[Funcs to DB, otherwise shld be a BE func]

[Which pg is it used for]

(likely FE side)

User Auth

- + register [Reg]
- + login [login]
- + logout [Profile]

Note: Firebase handles pw

User

- + get User [Profile]
- + add User [General]
- + update User [Profile]
- + remove User [Profile]

Product

- + getProduct
- + get Products [Product] [DB]

Product Supply [Product Listing]

- + get Listings By Product [DB]
- + calc Duration
 ↳ get wholesaler addr [API]
 ↳ get consumer addr
- + sort By Price
- + sort By Duration
- + sort By Stocks

Shopping Cart [Cart]

- + calc Total
- + view Items [DB]
- + add Item [DB]
- + update Item [DB]
- + delete Item [DB]

Payment [Payment]

- + view Mtds [DB]
- + add Mtd [DB]
- + delete Mtd [DB]
- + pay [DB]
 ↳ + convert Currency [API]

Order History [Order History]

- + add order [DB]
- + rate order [DB]
- + update order Status [DB]

