

accountType [ENUM]

- Admin (level 0)
- BuyerAndSeller (level 1)
- Seller (level 2)
- Buyer (level 3)

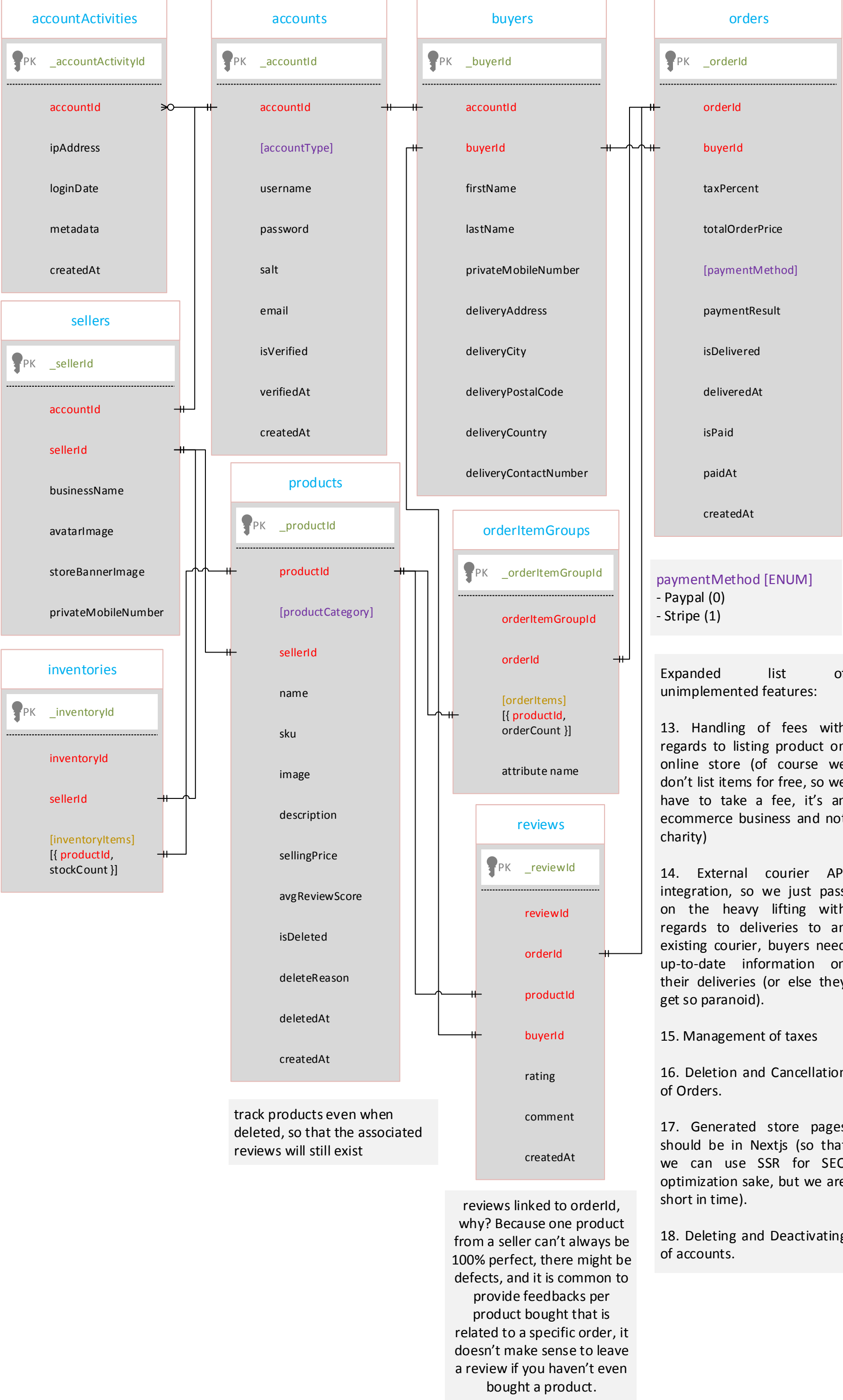
You cannot change your account type into a lower level, you cannot change your account from Buyer to Seller or vice-versa, you will only have the option to be both BuyerAndSeller (upgrading to level 1)

If there is at least one active order that is in a state of being processed, you cannot change your delivery address

RED = shown on public
GREEN = internal mongodb key.

To reduce complexity (and finish this portfolio quickly), did not implement the following features:

- Multiple Product Categories (& editing of available categories)
- Product Returns
- Chat/Messaging
- Two-Factor-Authentication
- Federated Security (sign up/sign in via Google or Facebook account)
- Multiple Product Images (cloudinary comes into mind for media hosting on cloud)
- Discount / Flash Sale Events
- SMS Notifications on order operations.
- Address Lookup via existing APIs to validate it's existence.
- Verification via SMS (email only, made possible by using captcha to mitigate spammers)
- Audit Trail, yep it's required to track changes for investigative purposes, but decided to skip it.
- Inventory SMS notifications of low stock level



paymentMethod [ENUM]

- Paypal (0)
- Stripe (1)

- Expanded list of unimplemented features:
- Handling of fees with regards to listing product on online store (of course we don't list items for free, so we have to take a fee, it's an ecommerce business and not charity)
 - External courier API integration, so we just pass on the heavy lifting with regards to deliveries to an existing courier, buyers need up-to-date information on their deliveries (or else they get so paranoid).
 - Management of taxes
 - Deletion and Cancellation of Orders.
 - Generated store pages should be in Nextjs (so that we can use SSR for SEO optimization sake, but we are short in time).
 - Deleting and Deactivating of accounts.

ProductService
<div>QUERY</div> <div>+getAllActiveProducts(): Promise<Product[]> +getProductById(productId: string): Promise<Product> +getAllProductsBySellerId(sellerId: string): Promise<Product[]> +getProductsByName(name: string): Promise<Product[]></div>
<div>MUTATION</div> <div>+createProduct(createProductInput: CreateProduct): Promise<Product> +updateProduct(updateProductInput: UpdateProduct): Promise<Product> +deleteProduct(productId: string): Promise<Product></div>

AccountService
<div>QUERY</div> <div>+getAccountById(accountId: string): Promise<Account></div>
<div>MUTATION</div> <div>+createAccount(createAccountInput: CreateAccount): Promise<Account> +updateProduct(updateProductInput: UpdateAccount): Promise<Account> +deleteProduct(productId: string): Promise<Product></div>