

AccountTypeName:

- Admin (level 4)
- BuyerAndSeller (level 3)
- Seller (level 2)
- Buyer (level 1)

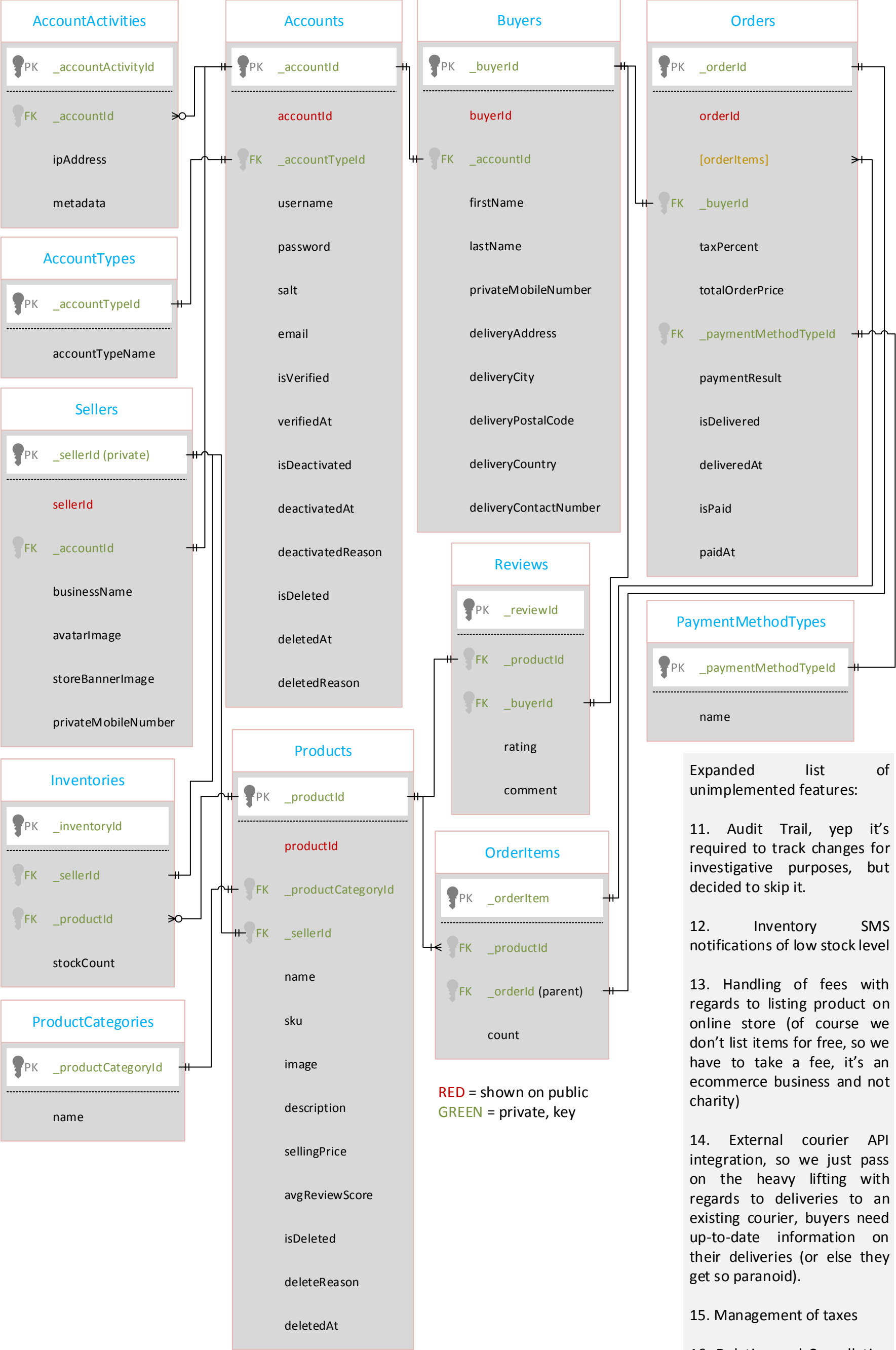
You cannot change your account type into a lower level, you cannot change your account from Buyer to Seller or vice-versa, you will only have the option to be both BuyerAndSeller (upgrading to level 3)

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

Why do we have public Ids? We don't want to expose the actual Ids used in the database (for security purposes), its fine to use UUID v4 for these. UUID v5 for private primary keys. What about collisions? Well, this is a mongodb database we'll gonna use, should have not included PKs and FKs as column classification, but it looks better to have those (for the sake of information). But yeah, I will worry about that if needed. This practice is prevalent in postgres.

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

1. Multiple Product Categories
2. Product Returns
3. Chat/Messaging
4. Two-Factor-Authentication
5. Federated Security (sign up/sign in via Google or Facebook account)
6. Multiple Product Images (cloudinary comes into mind for media hosting on cloud)
7. Discount / Flash Sale Events
8. SMS Notifications on order operations.
9. Address Lookup via existing APIs to validate it's existence.
10. Verification via SMS (email only, made possible by using captcha to mitigate spammers)



- Expanded list of unimplemented features:
11. Audit Trail, yep it's required to track changes for investigative purposes, but decided to skip it.
  12. Inventory SMS notifications of low stock level
  13. 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)
  14. 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).
  15. Management of taxes
  16. Deletion and Cancellation of Orders.
  17. Generated store pages should be in Nextjs (so that we can use SSR for SEO optimization sake, but we are short in time).

ProductService
<div>QUERY</div> <div>+findByProductId(_productId: string): Promise&lt;Product&gt;</div> <div>+findAllProductsBySellerId(_sellerId: string): Promise&lt;Product[]&gt;</div> <div>+findProductsByName(name: string): Promise&lt;Product[]&gt;</div>
<div>MUTATION</div> <div>+createProduct(createProductInput: CreateOrUpdateProduct): Promise&lt;Product&gt;</div> <div>+updateProduct(_productId: string, updateProductInput: CreateOrUpdateProduct): Promise&lt;Product&gt;</div> <div>+deleteProduct(_productId: string): Promise&lt;Product&gt;</div>