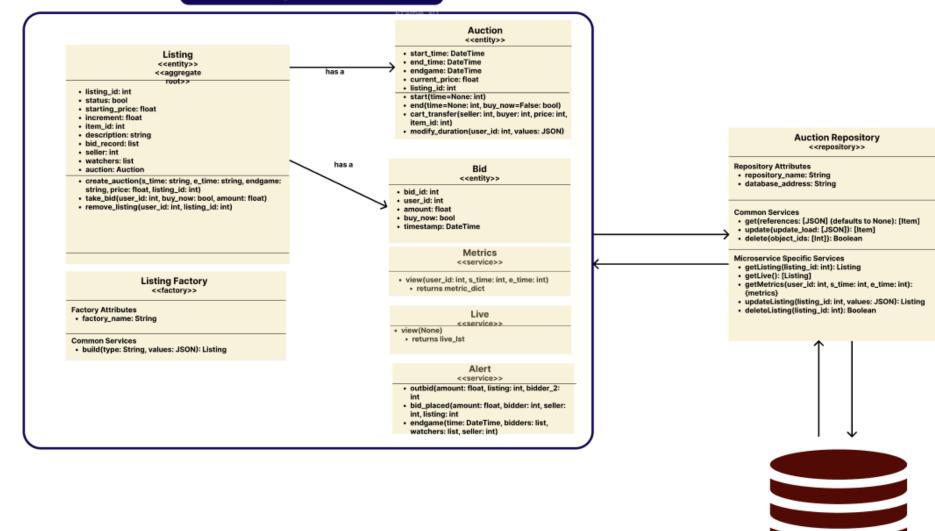
Flag Management Domain Model

Flag <<entity>> itemId : int userId : string flagld: int · reason:string Flag Repository <<repository>> view_flags(user_id=None: int, item_id=None: int): Flag review_flag(flag_id: int, user_id: int, response: bool): Repository Attributes · repository_name: String verdict: string · database_address: String Common Services get(references: [JSON] (defaults to None): [Item] update(update_load: [JSON]): [Item] delete(object_ids: [Int]): Boolean Microservice Specific Services getFlag(flag_id: int) getFlagger(user_id:int) getItem(item_id: int) removeFlag(user_id: int, flag_id: int)

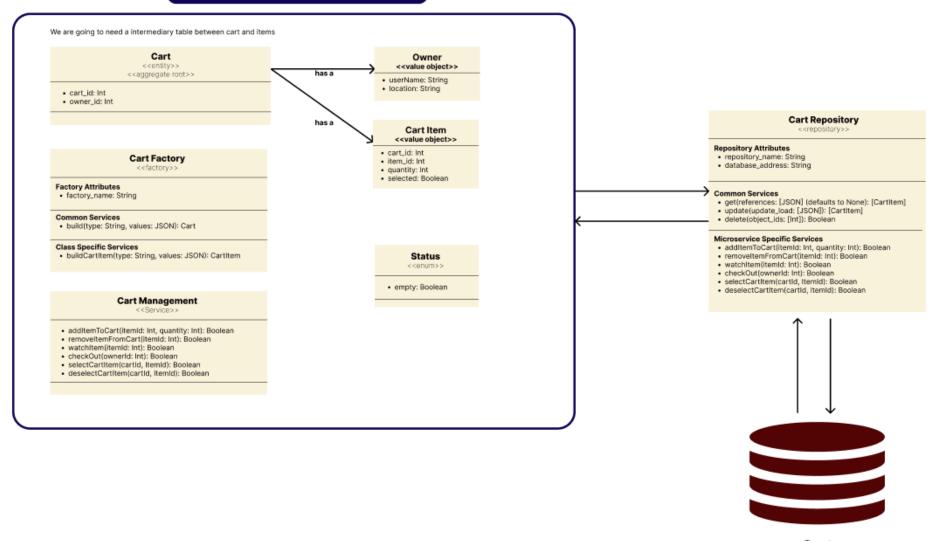
Flag Management Mongo Database

Auction Management Domain Model



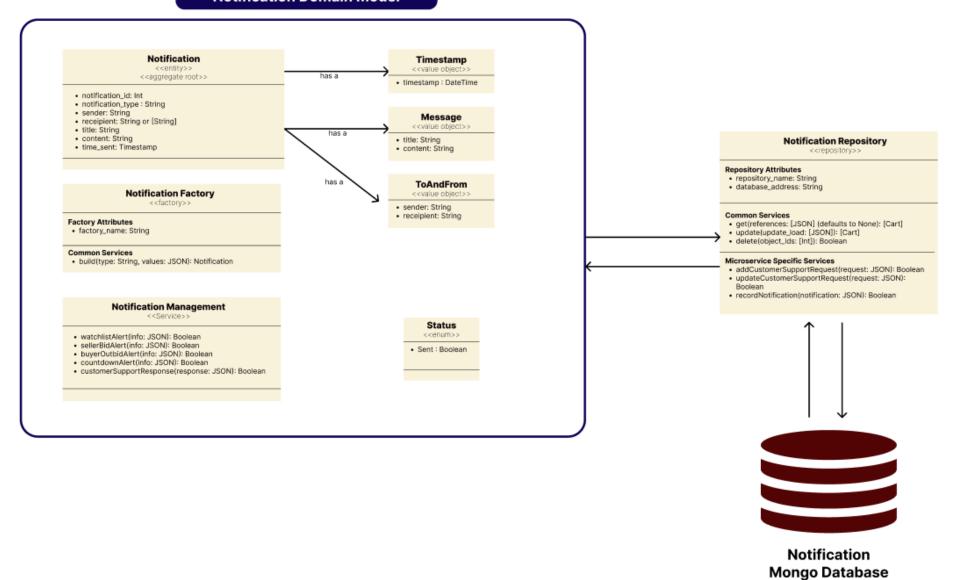
Auction Management Mongo Database

Cart Management Domain Model

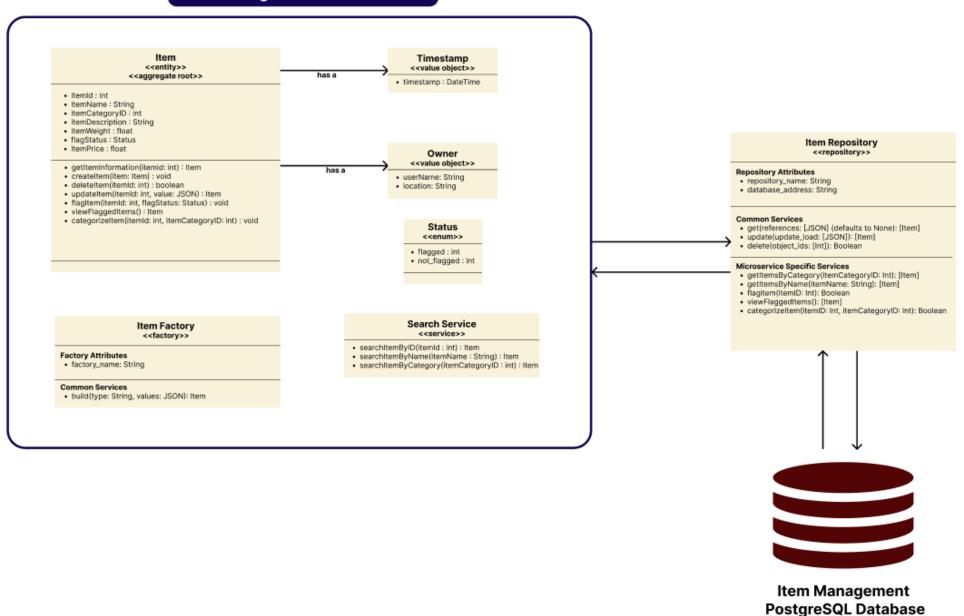


Cart PostgreSQL Database

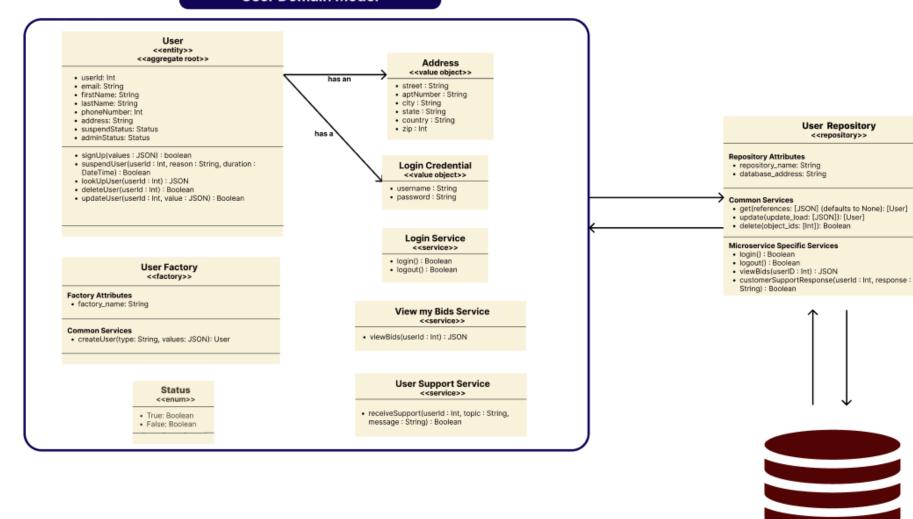
Notification Domain Model



Item Management Domain Model

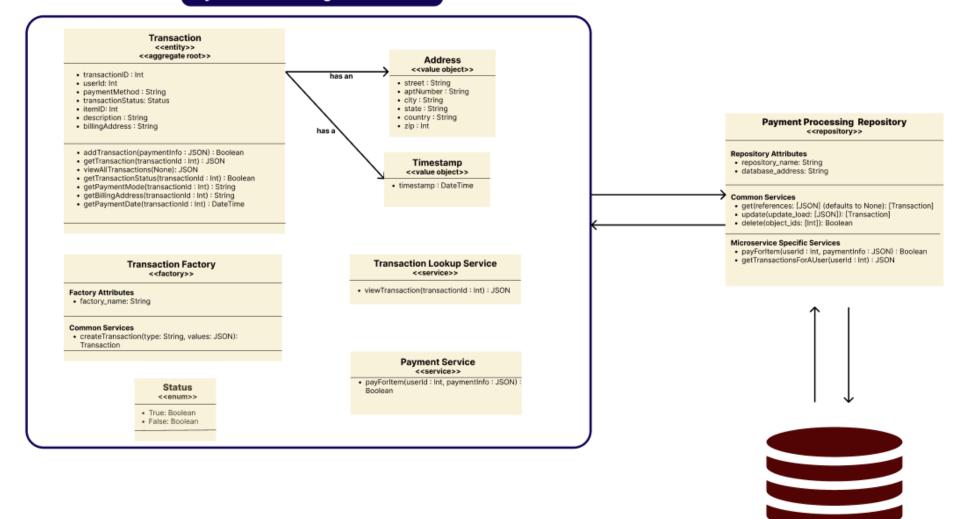


User Domain Model



User Management PostgreSQL Database

Payment Processing Domain Model



Payment Processing PostgreSQL Database

Category Management Domain Model

Category <<entity>> <<aggregate root>>

- categoryId : int
 categoryName : String
 categoryDescription : String
- · createCategory(category: Category) : void

- deleteCategory(categoryld: int) : boolean
 updateCategory(categoryld: int, value: JSON) : Item
 categorizeItem(itemId: int, itemCategoryID: int) : void

Category Factory <<factory>>

Factory Attributes
• factory_name: String

Common Services

· build(type: String, values: JSON): Category

Category Repository <<repository>>

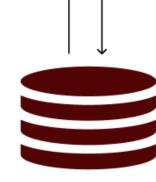
- Repository Attributes
 repository_name: String
- · database_address: String

Common Services

- get(references: [JSON] (defaults to None): [Category]
 update(update_load: [JSON]): (Category]
 delete(object_ids: [int]): Boolean

Microservice Specific Services

- getitemsByCategory(itemCategoryID: Int): [item]
 categorizeItem(itemID: Int, itemCategoryID: Int): Boolean



Category Management PostgreSQL Database