

SepTech Centauri Deliverable 4 Domain Model

***Disclaimer:** *This document is split up into Attribute and Association domain models separately to help reduce complexity of the diagram.*

Address
+ id: int + unit: str + building: str + street_name: str + street_type: str + city: str + region: str + country: str + address_code: str + postal_code: str
+ get_all(): Address[] + get_by_id(id: int): Address + update(id: int, updates: dict): Address + delete_by_id(id): Address + create(attrs: dict): Address

Bid
+ id: int + bid_amount: float + bid_posted_at: datetime
+ get_all(): Bid[] + get_by_id(id: int): Bid + update(id: int, updates: dict): Bid + delete_by_id(id): Bid + create(attrs: dict): Bid

BusinessBan
+ id: int + ban_start_date: datetime + ban_end_date: datetime + ban_reason: str
+ get_all(): BusinessBan[] + get_by_id(id: int): BusinessBan + update(id: int, updates: dict): BusinessBan + delete_by_id(id): BusinessBan + create(attrs: dict): BusinessBan

BusinessReport
+ id: int + report_reason: str + report_comment: str + created_at: datetime + reviewed_at: datetime
+ get_all(): BusinessReport[] + get_by_id(id: int): BusinessReport + update(id: int, updates: dict): BusinessReport + delete_by_id(id): BusinessReport + create(attrs: dict): BusinessReport

BusinessReview
+ id: int + created_at: datetime + rating: int + created_at: datetime + comment: str
+ get_all(): BusinessReview[] + get_by_id(id: int): BusinessReview + update(id: int, updates: dict): BusinessReview + delete_by_id(id): BusinessReview + create(attrs: dict): BusinessReview

Category
+ id: int + category1: str + category2: str + category3: str + category4: str + category5: str
+ get_all(): Category[] + get_by_id(id: int): Category + update(id: int, updates: dict): Category + delete_by_id(id): Category + create(attrs: dict): Category

Business
+ id: int + business_name: str + email: str + password: Binary(127) + owner_full_name: String(255) + created_at: DateTime + modified_at: DateTime + phone_number: String(11) + password_salt: String(63) + password_reset_code: String(6) + password_reset_timeout: DateTime + last_successful_login: DateTime + last_unsuccessful_login: DateTime + description: str + profile_picture: str(filepath)
+ get_all(): Business[] + get_by_id(id: int): Business + get_by_email(email: str): Business + update(id: int, updates: dict): Business + delete_by_id(id: int): Business + login(email: str, password: str): Business + reset_password(email: str): void - gen_unique_reset_code(): str + send_password_code_email(recipient: Business): void + verify_code(code: str, email: str): bool + check_exists(email: str): bool

ItemReport
+ id: int + report_reason: str + report_comment: str + created_at: datetime + reviewed_at: datetime
+ get_all(): ItemReport[] + get_by_id(id: int): ItemReport + update(id: int, updates: dict): ItemReport + delete_by_id(id): ItemReport + create(attrs: dict): ItemReport

ItemReviews
+ id: int + created_at: datetime + rating: int + created_at: datetime + comment: str
+ get_all(): ItemReviews[] + get_by_id(id: int): ItemReviews + update(id: int, updates: dict): ItemReviews + delete_by_id(id): ItemReviews + create(attrs: dict): ItemReviews

OrderItem
+ id: int
+ quantity: int
+ price: float*
+ added_at: datetime
+ get_all(): OrderItem[]
+ get_by_id(id: int): OrderItem
+ update(id: int, updates: dict): OrderItem
+ delete_by_id(id): OrderItem
+ create(attrs: dict): OrderItem

Order
+ id: int
+ ordered: bool
+ date_created: datetime
+ date_shipped: datetime
+ date_delivered: datetime
+ total_price: float*
+ get_all(): Order[]
+ get_by_id(id: int): Order
+ update(id: int, updates: dict): Order
+ delete_by_id(id): Order
+ create(attrs: dict): Order

RemovedItem
+ id: int
+ remove_date: datetime
+ remove_reason: str
+ get_all(): RemovedItem[]
+ get_by_id(id: int): RemovedItem
+ update(id: int, updates: dict): RemovedItem
+ delete_by_id(id): RemovedItem
+ create(attrs: dict): RemovedItem

Session
+ id: int
+ unique_session_id: str
+ session_expires_at: str
+ get_all(): Session[]
+ get_by_id(id: int): Session
+ update(id: int, updates: dict): Session
+ delete_by_id(id): Session
+ create(attrs: dict): Session

Subscription
+ id: int
+ tier: int
+ recurring: bool
+ subscription_ends: datetime
+ get_all(): Subscription[]
+ get_by_id(id: int): Subscription
+ update(id: int, updates: dict): Subscription
+ delete_by_id(id): Subscription
+ create(attrs: dict): Subscription

UserBan
+ id: int
+ ban_start_date: datetime
+ ban_end_date: datetime
+ ban_reason: str
+ get_all(): UserBan[]
+ get_by_id(id: int): UserBan
+ update(id: int, updates: dict): UserBan
+ delete_by_id(id): UserBan
+ create(attrs: dict): UserBan

UserReport
+ id: int
+ report_reason: str
+ report_comment: str
+ created_at: datetime
+ reviewed_at: datetime
+ get_all(): UserReport[]
+ get_by_id(id: int): UserReport
+ update(id: int, updates: dict): UserReport
+ delete_by_id(id): UserReport
+ create(attrs: dict): UserReport

Item
<ul style="list-style-type: none"> + id: int + name: str + quantity: str + created_at: datetime + modified_at: str + sellerid: int + can_buy: bool + buyout_price: float* + min_bid_increment: float* + starting_bid: float* + highest_bid: float* + bidding_ends: datetime + highest_bid_userid: int + quality: str + thumbnail: int + images: str[] + item_variants: int[] + description: str + attributes: json * stored as Numeric(10, 2) to preserve float values
<ul style="list-style-type: none"> + get_all(): Item[] + get_by_id(id: int): Item + update(id: int, updates: dict): Item + delete_by_id(id): Item + create(attrs: dict): Item + search(search_query: str): Item[]

User
<ul style="list-style-type: none"> + id: int + email: str + password: Binary(127) + full_name: String(255) + date_of_birth: Date + created_at: DateTime + modified_at: DateTime + phone_number: String(11) + password_salt: String(63) + password_reset_code: String(6) + password_reset_timeout: DateTime + last_successful_login: DateTime + last_unsuccessful_login: DateTime + admin_level: int
<ul style="list-style-type: none"> + get_all(): User[] + get_by_id(id: int): User + get_by_email(email: str): User + update(id: int, updates: dict): User + delete_by_id(id: int): User + create(attrs: dict): User + login(email: str, password: str): User + reset_password(email: str): void - gen_unique_reset_code(): str + send_password_code_email(recipient: User): void + verify_code(code: str, email: str): bool + check_exists(email: str): bool







