

WORKSHOP N° 2
SOFTWARE ENGINEERING II

Presented by:

Michael Stiven Betancourt Gelves

Camilo Andres Herrera Gutierrez

Cristhian Yamith Cely Oliveros

William David Pinilla Ocampo

Juan Manuel Cristancho Alvarez

Professor:

Carlos Andres Sierra Virguez

November 6th 2025



Universidad Nacional de Colombia
Engineering Faculty
2025

Software Requirements Specification - Click & Munch

CRC Cards

The following cards are the guide for the development of the application following the SOLID principles and defining the architecture of the system.

Auth Service:

Class: User

Responsibilities:

- Store user identity and credentials (id, name, email, password hash).
- Maintain role and basic profile data.
- Expose basic user info to other services.

Collaborators: Session, AuthService, ReservationService, OrderService

Class: Customer

Responsibilities:

- Inherit User attributes.
- Initiate reservations and pre-orders.
- Submit ratings and feedback

Collaborators: ReservationService, OrderService, Rating, NotificationService

Class: RestaurantManager

Responsibilities:

- Inherit User attributes.
- Manage restaurant profile and branch details.

- Propose menu changes and manage tables.

Collaborators: RestaurantService, MenuService, AdminService, NotificationService

Class: Waiter

Responsibilities:

- Inherit User attributes.
- View ready orders and mark service complete.
- Receive targeted notifications for deliveries.

Collaborators: OrderService, NotificationService, RestaurantService

Class: Chef

Responsibilities:

- Inherit User attributes.
- View kitchen order queue.
- Update dish and order statuses (Preparing, Ready, Served).

Collaborators: OrderService, MenuService, NotificationService

Class: SystemAdministrator

Responsibilities:

- Inherit User attributes.
- Approve or reject restaurant and menu requests.
- Generate and access system reports and admin functions.

Collaborators: AdminService, RestaurantService, MenuService

Class: Session

Responsibilities:

- Maintain active session state (JWT token, expiry).

- Validate token and refresh when required.

Collaborators: AuthService, JwtTokenUtil, User

Restaurant Service

Class: Restaurant

Responsibilities:

- Store core restaurant data (name, description, ratings summary).
- Expose restaurant metadata and links to branches & menus.

Collaborators: Branch, Menu, Rating, ReservationService, GeoService

Class: Branch

Responsibilities:

- Represent a single physical location (address, coordinates, contact).
- Store operating hours and branch-specific tables.

Collaborators: Restaurant, Table, OperatingHours, GeoService

Class: Table

Responsibilities:

- Represent a seating unit with capacity and current state.
- Track reservations and occupancy status.

Collaborators: ReservationService, Branch, Restaurant

Class: OperatingHours

Responsibilities:

- Define open/close schedules and exceptions (holidays).
- Provide availability checks for a given datetime.

Collaborators: Branch, ReservationService

Class: Rating

Responsibilities:

- Store individual rating records and comments.
- Provide aggregated rating metrics for restaurant listing.

Collaborators: Customer, Restaurant, AdminService

Menu Service

Class: Menu

Responsibilities:

- Aggregate MenuItem's for a restaurant or branch.
- Provide category grouping and availability filters.

Collaborators: MenuCategory, MenuItem, RestaurantService

Class: MenuCategory

Responsibilities:

- Categorize menu items (e.g., Starters, Mains, Drinks).
- Support category-level operations (visibility, ordering).

Collaborators: Menu, MenuItem

Class: MenuItem

Responsibilities:

- Define dish properties: name, description, price, ETA, availability.
- Provide variants and customization options.

Collaborators: MenuCategory, CustomizationOption, OrderService

Class: CustomizationOption

Responsibilities:

- Represent configurable options (size, modifiers, add-ons).
- Validate option compatibility with MenuItem.

Collaborators: MenuItem, OrderItem

Reservation Service

Class: Reservation

Responsibilities:

- Represent a booking: customer, branch, table(s), datetime, party size, status.
- Enforce reservation business rules and persist history.

Collaborators: Customer, Branch, Table, ReservationStatus, NotificationService

Class: ReservationStatus

Responsibilities:

- Enumerate and represent reservation states (Pending, Confirmed, Cancelled, Completed).

Collaborators: Reservation

Class: Confirmation

Responsibilities:

- Capture confirmation interactions (e.g., push confirm/decline).
- Trigger follow-up actions (notify restaurant, update reservation).

Collaborators: Reservation, NotificationService

Class: CancellationPolicy

Responsibilities:

- Define rules for cancellation (deadlines, penalties if any).
- Validate cancellation requests against policy.

Collaborators: Reservation, AdminService

Order Service

Class: Order

Responsibilities:

- Represent an order tied to a reservation or table.
- Maintain order lifecycle and total cost.

Collaborators: OrderItem, Customer, Chef, Waiter, NotificationService

Class: OrderItem

Responsibilities:

- Represent one dish within an Order: menu item ref, quantity, notes, chosen customizations.
- Calculate line cost.

Collaborators: Order, MenuItem, CustomizationOption

Class: OrderStatus

Responsibilities:

- Enumerate order lifecycles (Created, Preparing, Ready, Served, Cancelled).

Collaborators: Order

Class: PreparationTime

Responsibilities:

- Store or compute estimated preparation time per menu item or order.
- Provide ETA for order or individual items.

Collaborators: MenuItem, Order

Class: Tip

Responsibilities:

- Record optional gratuity amounts and associate with the waiter or order.

Collaborators: Order, Waiter

Notification Service

Class: Notification

Responsibilities:

- Represent and persist a notification event (recipient, payload, type, delivered flag).
- Route notifications to the proper delivery channel.

Collaborators: User, NotificationType, Channel, NotificationPreference

Class: NotificationType

Responsibilities:

- Define categories: ReservationReminder, ReservationConfirmed, OrderReady, OrderStatusUpdate, AdminMessage.

Collaborators: Notification, NotificationPreference

Class: NotificationPreference

Responsibilities:

- Store user preferences for receiving notifications (channels, snooze settings).

Collaborators: User, Notification

Class: Channel

Responsibilities:

- Abstract delivery channels: Push (FCM), Email, SMS, WebSocket.

- Provide send and retry logic for each channel.

Collaborators: Notification, NotificationService

Admin Service

Class: RestaurantRequest

Responsibilities:

- Represent a restaurant signup/request record for admin review.
- Track approval state and admin comments.

Collaborators: Restaurant, SystemAdministrator, NotificationService

Class: MenuRequest

Responsibilities:

- Represent a submitted menu or menu change pending approval.
- Record audit trail and versioning information.

Collaborators: Menu, SystemAdministrator, NotificationService

Class: Report

Responsibilities:

- Generate aggregations and exports (reservations, orders, ratings, activity).
- Provide API for CSV/PDF exports.

Collaborators: AdminService, ReservationService, OrderService, RestaurantService

Class: SystemUser

Responsibilities:

- Represent internal admin users and their roles/permissions.
- Facilitate audit and action logging.

Collaborators: SystemAdministrator, AdminService

Geo Service

Class: Location

Responsibilities:

- Store coordinates (latitude, longitude) and location metadata.
- Provide helper functions for distance calculations.

Collaborators: Branch, Customer, Geo queries (PostGIS)

Class: SearchFilter

Responsibilities:

- Encapsulate search parameters (radius, cuisine, rating threshold, sort order).

Collaborators: GeoService, RestaurantService

Class: SearchResult

Responsibilities:

- Contain ordered list of nearby restaurants with distance and ranking metadata.

Collaborators: GeoService, Restaurant

Shared Library (Common)

Class: BaseEntity

Responsibilities:

- Provide common persistence fields (id, createdAt, updatedAt, version).

Collaborators: All domain entities

Class: ApiResponse

Responsibilities:

- Standardize successful API responses (status, data, meta).

Collaborators: All Services, Controllers

Class: ErrorResponse

Responsibilities:

- Standardize error payloads (code, message, details).

Collaborators: Exception handlers across services

Class: JwtTokenUtil

Responsibilities:

- Generate, sign and validate JWT tokens.
- Extract claims and validate expiration/audience.

Collaborators: AuthService, Session

Class: MapperUtil

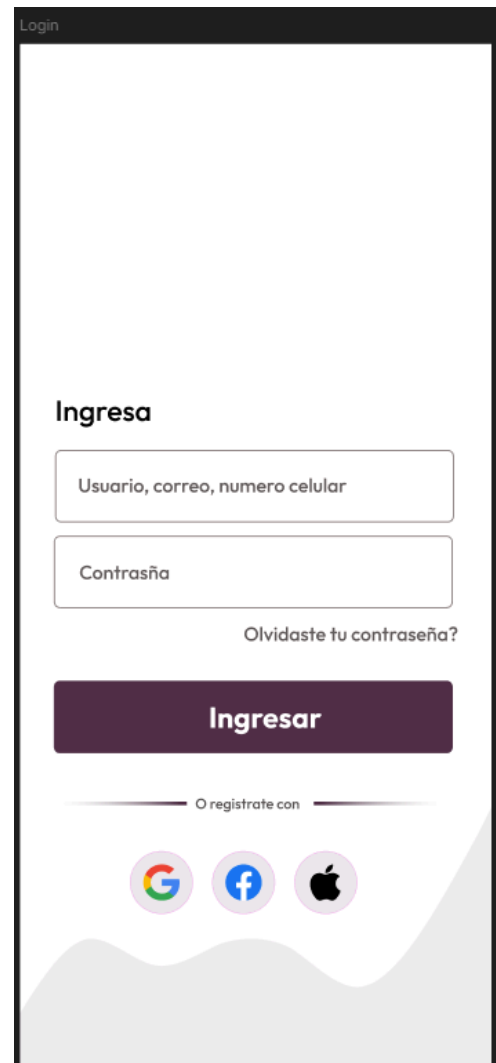
Responsibilities:

- Convert Entities ↔ DTOs and validate payload mappings.
- Centralize mapping rules to avoid duplication.

Collaborators: Controllers, Services, Repositories

Mockups

<https://www.figma.com/design/OsUJLRGwAVkChvzjCQdMFN/MockupsClick-Munch?node-id=0-1&t=SnMYEkC37pTQu0pO-1>



📍 Bogotá
Colombia



🔍 Busca el plato o restaurante

**40% de
descuento**

EN TU PRIMERA ORDEN

RESERVA YA!

★ Populares

[Ver Mas](#)



changua



tamal



paiza



hamburg

★ El especial de hoy

[Ver Mas](#)



Cocido boyasence

20.000 ~~\$30.000~~

20%

Amet minim mollit non deserunt ullamco est sit aliqua dolor do amet sint.

★ 5.0(34) Principal

RECOMENDADO



Arroz con

\$26.000

Amet minim

★ 5.0(34)

★ El especial de hoy

[Ver Mas](#)



Cocido boyasence

20.000 ~~\$30.000~~

20%

Amet minim mollit non deserunt ullamco est sit aliqua dolor do amet sint.

★ 5.0(34) Principal

RECOMENDADO



Arroz con

\$26.000

Amet minim

★ 5.0(34)

★ Deliciosos postres

[Ver Mas](#)



Miloja

5.000 ~~\$8.000~~

20%

Amet minim mollit non deserunt ullamco est sit aliqua dolor do amet sint.

★ 5.0(34) Principal

RECOMENDADO



Fresas

\$9.999

Amet minim

★ 5.0(34)



Inicio



Menu



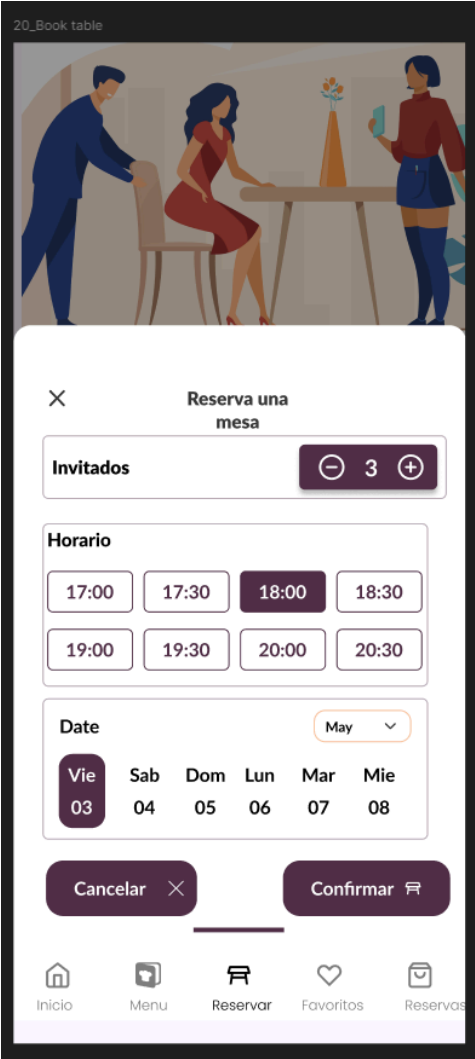
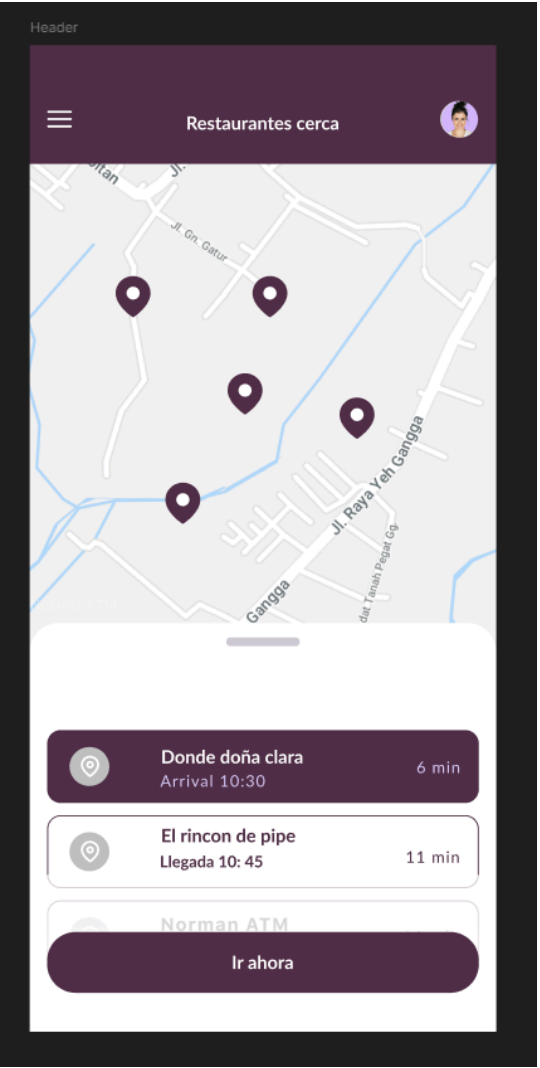
Reserva

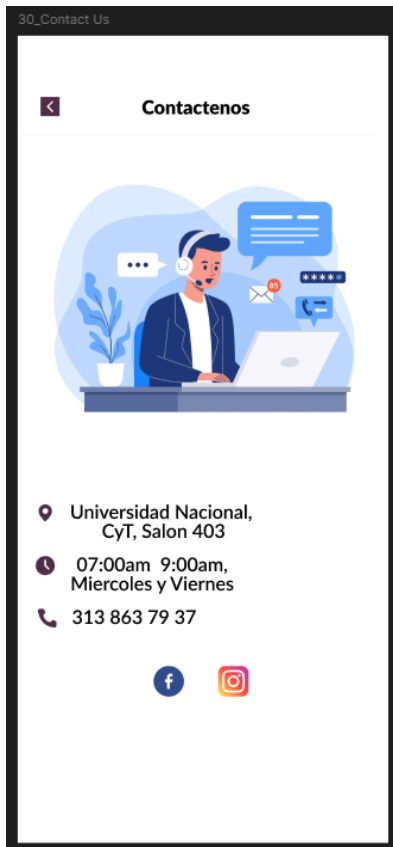


Favorito



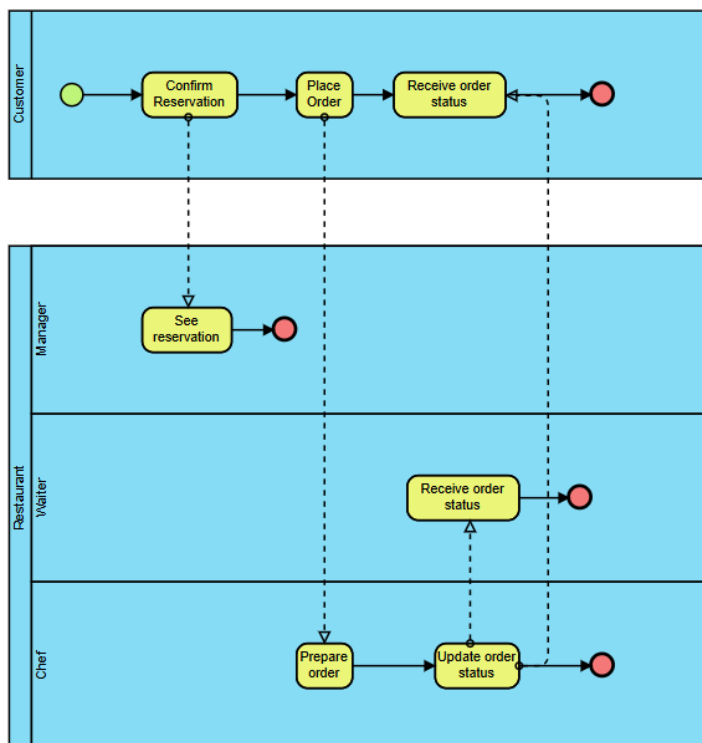
Carrito



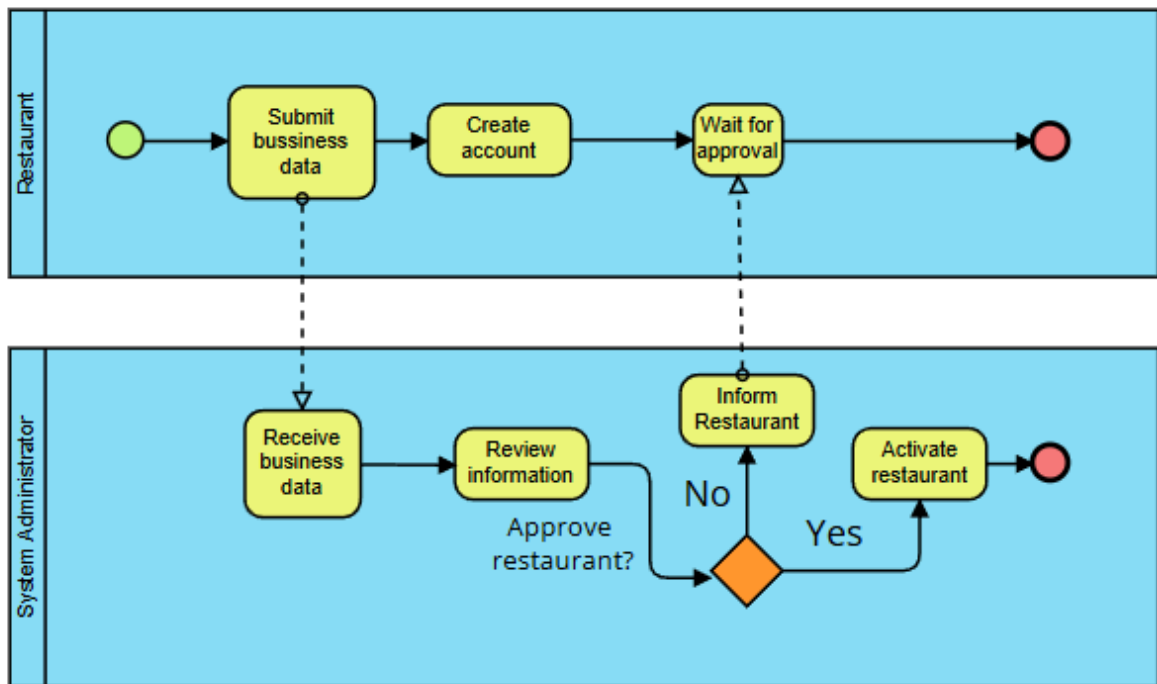


Business Model Process: BPMN Diagram

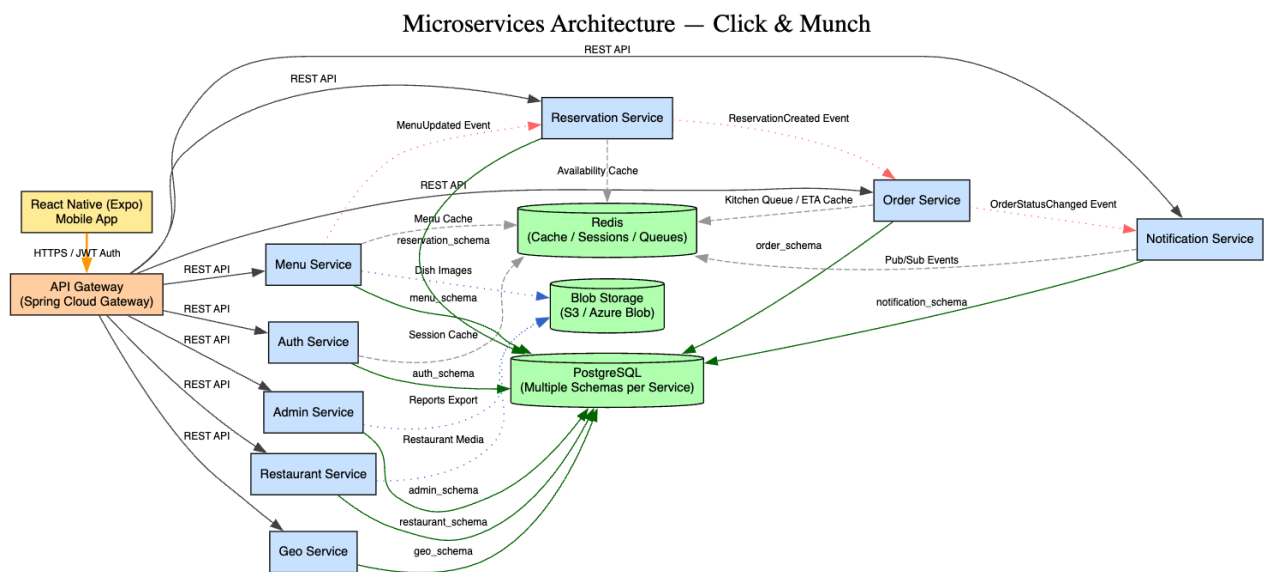
Make a reservation and place and order



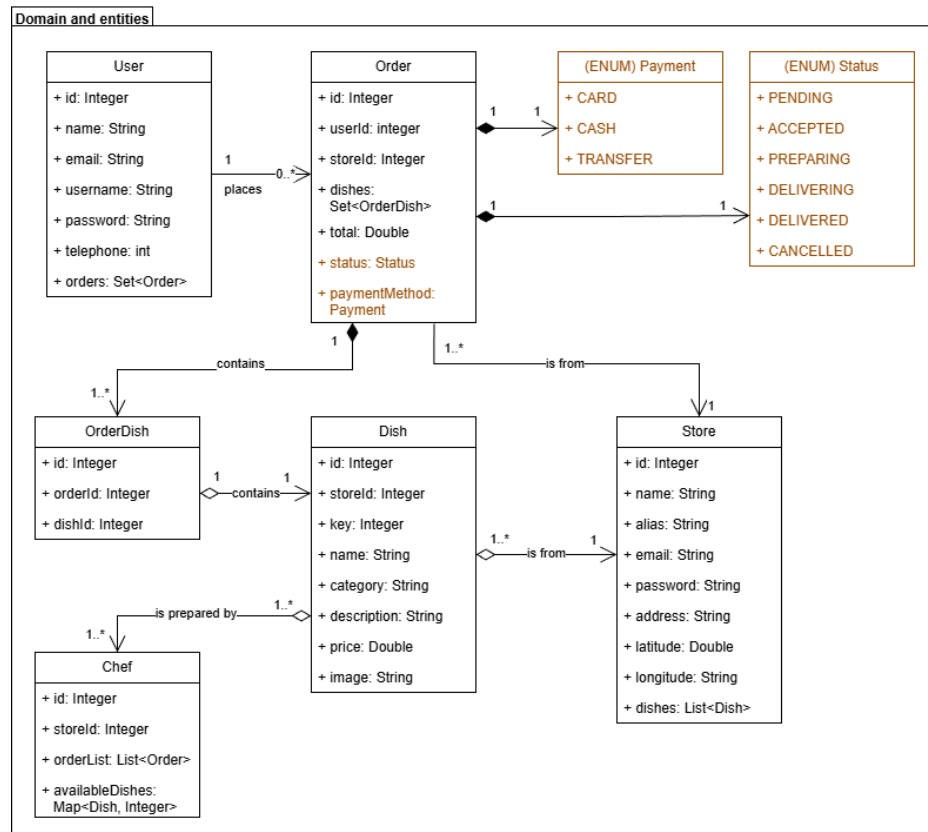
Restaurant registration



Architecture Diagram

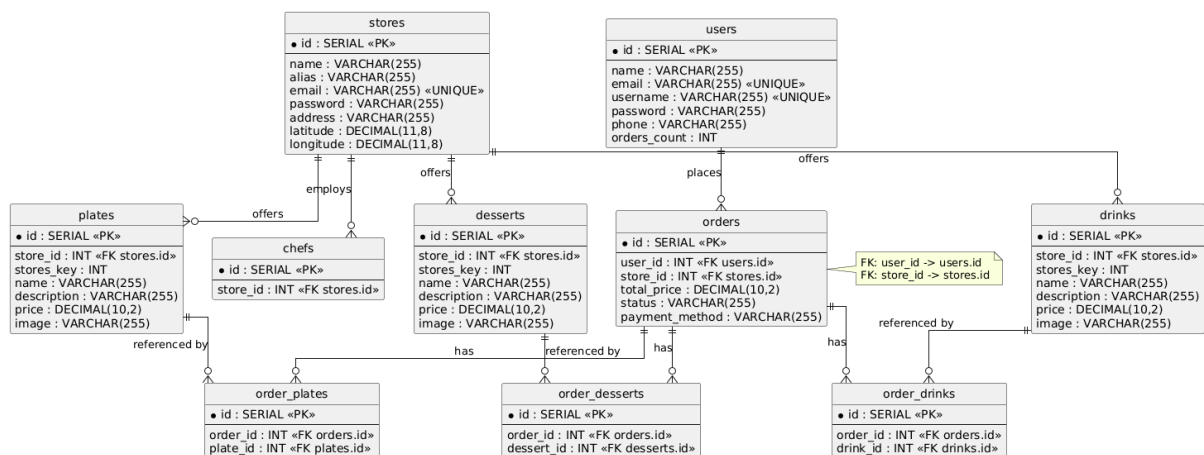


UML Class Diagram for domain/entities within the backend application:



UML Class Diagram for the backend application microservices:

Relational Database Model



References

EngAndres. (n.d.). *unal_public* [Folder: Software Engineering 2_Morning (G3)/slides]. GitHub.
[https://github.com/EngAndres/unal_public/tree/main/Software%20Engineering%202_Morning%20\(G3\)/slides](https://github.com/EngAndres/unal_public/tree/main/Software%20Engineering%202_Morning%20(G3)/slides)