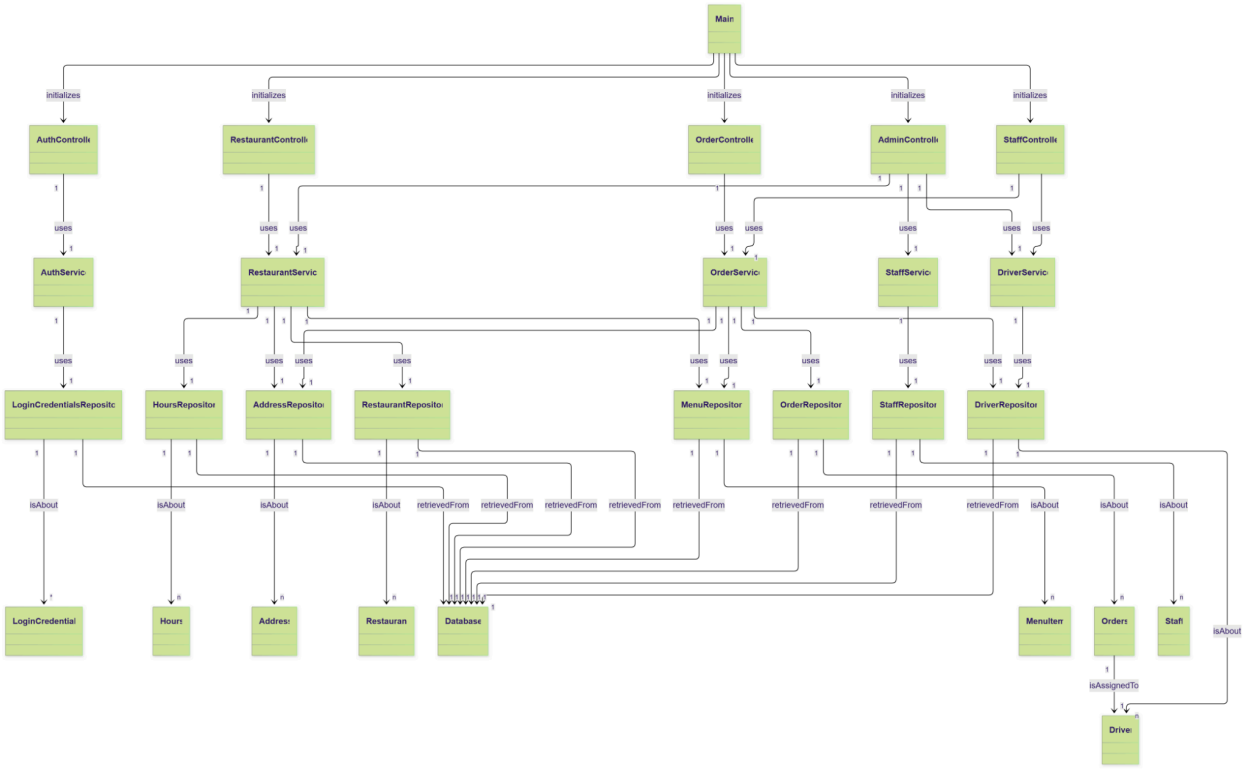
Rubyanne Moley, Mollie Hamman, Luke Stanford

CS 5336/7336 – Web Application Development Project Title: FrontDash Application Design Document

Class Diagram



The diagram follows a layered architecture consisting of Controllers, Services, Repositories, Models, and a Database access layer. Each class is shown with its key attributes and methods, and the relationships between classes illustrate how data flows through the system. Controllers receive HTTP requests from the front end, Services contain the business logic, Repositories handle data persistence, and Model classes represent application entities.

This UML class diagram captures the complete set of back-end components used to manage restaurants, orders, staff, drivers, credentials, menus, and hours of operation. Multiplicities are included to indicate the structure of associations, and method names reflect the operations required for the application's core features. This design serves as the foundation for the final implementation of the application's server-side functionality.

Rubyanne Moley, Mollie Hamman, Luke Stanford

Detailed Description

Classname: Restaurant
Attributes:
restName : String – Unique restaurant identifier.
picture : String – URL or path to the restaurant's picture.
contactName : String – Main point of contact for the restaurant.
contactEmail : String – Email of the restaurant contact.
contactPhone : String – Phone number of the restaurant contact.
isActive : Boolean – Indicates whether the restaurant is approved and active.
pendingWithdraw : Boolean – True if the restaurant has requested a withdrawal.
isAvailable : Boolean – True if the restaurant is accepting orders.

Classname: Orders
Attributes:
orderNumber : int – Unique order ID.
restName : String – Restaurant fulfilling the order.
driverName : String – Driver assigned to the order.
orderDate : Date – Date the order was created.
orderTime : Time – Time the order was created.
subtotalAmount : number – Cost of all menu items.
serviceCharge : number – Platform fee.
tipAmount : number – Tip for the driver.
grandTotal : number – Complete total including fees and tip.
orderStatus : String – Status such as pending, assigned, delivered.
deliveryDate : Date – Date the order was delivered.
deliveryTime : Time – Time the order was delivered.

Classname: Driver
Attributes:
driverName : String – Unique driver identifier.
employmentStatus : String – Employment status.
isAvailable : String – Tracks if the driver is free for assignment.

Classname: Staff
Attributes:
username : String – Login username.
firstName : String – Staff first name.
lastName : String – Staff last name.
employmentStatus : String – Active or inactive staff status.

Classname: LoginCredentials
Attributes:
username : String – Username used for login.
password : String – Hashed password.

Rubyanne Moley, Mollie Hamman, Luke Stanford

userType : String – Role category (staff or restaurant).

Classname: MenuItem
Attributes:
itemId : int – Unique menu item ID.
restName : String – Restaurant that owns this item.
itemName : String – Name of the dish.
price : number – Price of the item.
isAvailable : Boolean – Whether the item can be ordered.

Classname: Hours
Attributes:
restName : String – Restaurant these hours belong to.
dayOfWeek : String – Day of the week.
openTime : Time – Time the restaurant opens.
closeTime : Time – Time the restaurant closes.

Classname: Address
Attributes:
street1 : String – Primary street line.
street2 : String – Secondary line.
city : String – City.
state : String – State.
zip : String – Postal code.

Classname: RestaurantRepository
Methods:
findByName(restName)
createRegistration(data)
setApprovalStatus(restName, status)
requestWithdrawal(restName)
adminSetWithdrawal(restName, decision)
listRestaurants(includeInactive, includePendingWithdraw)

Classname: DriverRepository
Methods:
createDriver(name)
setStatus(name, status)
assignDriver(orderNumber, driverName)
listDrivers()

Classname: StaffRepository
Methods:
createStaff(data)

Rubyanne Moley, Mollie Hamman, Luke Stanford

setStatus(username, status)
listStaff()

Classname: LoginCredentialsRepository
Methods:
findByUsername(username)
verifyStaffLogin(username, password)
verifyRestaurantLogin(username, password)

Classname: MenuRepository
Methods:
getMenuByRestaurant(restName)
updateMenuItem(restName, itemId, data)

Classname: HoursRepository
Methods:
updateHours(restName, dayOfWeek, data)

Classname: AddressRepository
Methods:
createOrFindAddress(street1, street2, city, state, zip)

Classname: OrderRepository
Methods:
createOrder(restName)
getOrderSummary(orderNumber)
setDeliveryTime(orderNumber, date, time)
addItem(orderNumber, itemId, qty)
listOrders()

Classname: AuthService
Methods:
loginStaff(username, password)
loginRestaurant(username, password)

Classname: RestaurantService
Methods:
requestRegistration(data)
adminApproveRestaurant(restName)
adminRejectRestaurant(restName)
requestWithdrawal(restName)
adminHandleWithdrawal(restName, decision)
updateMenuItem(restName, itemId, data)
updateHours(restName, dayOfWeek, data)

Rubyanne Moley, Mollie Hamman, Luke Stanford

Classname: OrderService
Methods:
createOrder(restName)
getOrderSummary(orderNumber)
addItemToOrder(orderNumber, itemId, qty)
assignDriverToOrder(orderNumber, driverName)
markOrderDelivered(orderNumber)
listOrders()

Classname: DriverService
Methods:
createDriver(name)
setDriverStatus(name, status)
listDrivers()

Classname: StaffService
Methods:
createStaff(data)
setStaffStatus(username, status)
listStaff()

Classname: AuthController
Methods:
postLoginStaff(req, res)
postLoginRestaurant(req, res)

Classname: RestaurantController
Methods:
postRegisterRestaurant(req, res)
putMenuItem(req, res)
putHours(req, res)
postWithdrawal(req, res)

Classname: AdminController
Methods:
putRestaurantApproval(req, res)
putRestaurantWithdrawal(req, res)
getRestaurants(req, res)
postStaff(req, res)
putStaffStatus(req, res)
getStaff(req, res)
postDriver(req, res)
putDriverStatus(req, res)

getDrivers(req, res)

Classname: StaffController
Methods:
putOrderDriver(req, res)
putOrderDelivered(req, res)

Classname: OrderController
Methods:
postOrder(req, res)
getOrder(req, res)
postOrderItem(req, res)
putOrderDelivery(req, res)
getOrders(req, res)

Classname: Database
Methods:
query(sql, params)
callStoredProcedure(name, params)

Classname: Main
Purpose:
Initializes controllers and services.

Springboot Architecture

Models

Classname	**Setter/Update Methods**
Restaurant	setActiveStatus, setPendingWithdraw, setAvailability, updateContactInfo
Orders	updateStatus, updateDeliveryTime
Driver	updateEmploymentStatus, updateAvailability
Staff	updateEmploymentStatus
LoginCredentials	(no special setters beyond basic field setters)
MenuItem	updatePrice, updateAvailability
Hours	updateOpenCloseTimes

Address	updateAddressLines, updateCityStateZip

Service Classes

Classname	**Setter/Update Methods**
AuthService	loginStaff, loginRestaurant
RestaurantService	requestRegistration, adminApproveRestaurant, adminRejectRestaurant, requestWithdrawal, adminHandleWithdrawal, updateMenuItem, updateHours
OrderService	createOrder, getOrderSummary, addItemToOrder, assignDriverToOrder, markOrderDelivered, listOrders
DriverService	createDriver, setDriverStatus, listDrivers
StaffService	createStaff, setStaffStatus, listStaff

Repository Classes

Classname	**Setter/Update Methods**
RestaurantRepository	findByName, createRegistration, setApprovalStatus, requestWithdrawal, adminSetWithdrawal, listRestaurants
OrderRepository	createOrder, getOrderSummary, setDeliveryTime, addItem, listOrders
DriverRepository	createDriver, setStatus, listDrivers
StaffRepository	createStaff, setStatus, listStaff
LoginCredentialsRepository	findByUsername, verifyStaffLogin, verifyRestaurantLogin
MenuRepository	getMenuByRestaurant, updateMenuItem
HoursRepository	updateHours

AddressRepository	createOrFindAddress

Controller

AuthController – handles staff and restaurant login requests using AuthService.
RestaurantController – handles restaurant registration, menu updates, hours updates, and withdrawal requests using RestaurantService.
AdminController – handles admin approval/rejection of restaurant registrations, handles restaurant withdrawal decisions, and manages staff and driver accounts (create, update status, list) using RestaurantService, StaffService, and DriverService.
OrderController – handles creating orders, adding items, listing orders, and retrieving an order summary using OrderService.
StaffController – handles assigning drivers to orders and marking orders as delivered using OrderService and DriverService.