ParkZone UML Class Diagram

Admin
```
+ managedSlots : List<ParkingSlot>

+ managedSlots : List<ParkingSlot>
+ addParkingSlot(slot : ParkingSlot) : void
+ removeParkingSlot(slotID : int) : void
+ viewAllActiveTickets() : void
+ viewVehicleHistory(vehiclePlate : String) : void
+ getManagedSlots() : List<ParkingSlot>
+ addManagedSlots() : ParkingSlot
+ removeManagedSlot(int : int) : boolean
+ setHourlyRate(rate : double) : void
+ markSpaceOutOfService(slotID : int) : void
+ returnSpaceToService(slotID : int) : void
```

ClientHandler
```
- server : ParkingSystemServer
- socket : Socket
- clientId : String
- in : ObjectInputStream
- out : ObjectOutputStream
- open : AtomicBoolean
- running : boolean
- loggedIn : boolean
- currentUser : User
- parkingSystem : ParkingSystem
- ticketService : TicketService

+ ClientHandler(server : ParkingSystemServer, socket : Socket)
+ run() : void
+ readLoop() : void
+ handleMessage(msg : Message) : void
+ handleLogin(msg : Message) : void
+ handleLogout(msg : Message) : void
+ handleAddSlots(msg : Message) : void
+ handleGetTickets(msg : Message) : void
+ handleRemoveSlot(msg : Message) : void
+ handleReserveSlot(msg : Message) : void
+ authenticateIfRequired(msg : Message) : boolean
+ send(msg : Message) : void
+ onServerPush(msg : Message) : void
+ findSlotById(slotId : int) : ParkingSlot
+ computeNextSlotIdFromServer() : int
+ close() : void
+ cleanup() : void
```

<<Interface>>
MessageCodec
```
+ field: type

+ read(in : InputStream) : Object
+ write(out : OutputStream, msg : Object) : void
```

Message
```
- type : String
- status : String
- text : String
- payload : Object

+ Message(type : String)
+ getType() : String
+ setType(type : String) : void
+ getStatus() : String
+ setStatus(status : String) : void
+ getText() : String
+ setText(text : String) : void
+ getPayload() : Object
+ setPayload(payload : Object) : void
+ static login(username : String) : Message
+ static text(content : String) : Message
```

<<Interface>>
Actionable
```
+ getActionName() : String
+ execute() : void
```

ParkingSystemServer
```
- port : int
- serverSocket : ServerSocket
- clientPool : ExecutorService
- running : AtomicBoolean
- clientsById : Map<String, ClientHandler>
- parkingSystem : ParkingSystem
- ticketService : TicketService

+ ParkingSystemServer(port : int)
+ start() : void
+ stop() : void
+ acceptLoop() : void
+ newClientHandler(socket : Socket) : ClientHandler
+ removeClient(clientId : String) : void
+ broadcast(msg : Message) : void
+ sendTo(clientId : String, msg : Message) : void
+ getParkingSystem() : ParkingSystem
+ getTicketService() : TicketService
+ main(args : String[]) : void
```

<<Abstract>>
User
```
+ userID : int
+ firstName : String
+ lastName : String
+ email : String
+ password : String
+ accountType : String
- actions : Actionable[]

+ login(email, password) : boolean
+ logout() : void
+ getUserID() : int
+ setUserID(id : int) : void
+ getFirstName() : String
+ setFirstName(n : String) : void
+ getLastName() : String
+ setLastName(n : String) : void
+ getEmail() : String
+ setEmail(e : String) : void
+ getPassword() : String
+ setPassword(pw : String) : void
+ getAccountType() : String
+ setAccountType(t : String) : void
```

ParkingGarage
```
- garageID : int
- name : String
- address : String
- hourlyRate : double

+ getSlotFor(typeHint : String) : ParkingSlot
+ getGarageID() : int
+ setGarageID(gid : int) : void
+ getName() : String
+ setName(n : String) : void
+ getAddress() : String
+ setAddress(a : String) : void
+ getHourlyRate() : double
+ setHourlyRate(hr : double) : void
```

<<Singleton>>
ParkingSystem
```
- static instance : ParkingSystem
- users : List<User>
- slots : List<ParkingSlot>
- tickets : List<Ticket>
- ticketService : TicketService
- slotCatalog : SlotCatalog

+ ParkingSystem()
+ static getInstance() : ParkingSystem
+ createAccount(user : User) : void
+ addUser(user : User) : void
+ addSlot(slot : ParkingSlot) : void
+ addTicket(ticket : Ticket) : void
+ removeSlotById(slotId : int) : boolean
+ login(email : String, password : String) : User
+ registerVehicle(client : Client, v : Vehicle) : void
+ issueTicket(v : Vehicle, slot : ParkingSlot) : Ticket
+ endParking(ticketId : int) : void
+ closeTicket(ticketId : int, exitTime : LocalDateTime) : void
+ getUsers() : List<User>
+ getSlots() : List<ParkingSlot>
+ getTickets() : List<Ticket>
+ getSlotsSnapshot() : List<ParkingSlot>
+ getTicketsSnapshot() : List<Ticket>
+ findTicketById(ticketId : int) : Ticket
+ persistSystemState() : void
+ generateWeeklyReports() : void
+ runOverstayChecks() : void
+ broadcastSpaceUpdates() : void
```

<<Interface>>
SpaceUpdatePublisher
```
+ broadcastSpaceUpdates() : void
+ publishSlotUpdate(slotId : int) : void
+ publishGarageSummary(garageId : int) : void
+ publishOverstayAlerts() : void
```

<<Interface>>
SpaceUpdatePublisher
```
+ broadcastSpaceUpdates() : void
+ publishSlotUpdate(slotId : int) : void
+ publishGarageSummary(garageId : int) : void
+ publishOverstayAlerts() : void
```

TimeScheduler
```
+ triggerPersistSystemState() : void
+ triggerWeeklyReports() : void
+ triggerOverstayChecks() : void
+ triggerBroadcastSpaceUpdates() : void
```

Operator
```
+ monitorGarage() : void
+ viewActiveReservations() : List<Ticket>
+ processPaymentOnBehalf(t : Ticket, method : String) : boolean
+ resolveParkingIssue(ticketID : int, note : String) : void
+ markSpaceOutOfService(slotID : int) : void
+ returnSpaceToService(slotID : int) : void
+ viewDashboard() : void
```

SlotCatalog
```
- parkingSystem : ParkingSystem

+ findAvailable(garageID : int, typeHint : String) : List<ParkingSlot>
+ reserve(slotID : int) : void
+ release(slotID : int) : void
```

Ticket
```
- ticketId : int
- vehicle : Vehicle
- slot : ParkingSlot
- entryTime : LocalDateTime
- exitTime : LocalDateTime
- totalFee : double
- active : boolean
- reservedMinutes : int
- estimatedFee : double
- paid : boolean
- paymentMethod : String

+ Ticket(ticketId : int, vehicle : Vehicle, slot : ParkingSlot, entryTime : LocalDateTime)
+ getTicketId() : int
+ setTicketId(id : int) : void
+ getVehicle() : Vehicle
+ setVehicle(v : Vehicle) : void
+ getSlot() : ParkingSlot
+ setSlot(slot : ParkingSlot) : void
+ getEntryTime() : LocalDateTime
+ setEntryTime(time : LocalDateTime) : void
+ getExitTime() : LocalDateTime
+ setExitTime(time : LocalDateTime) : void
+ getTotalFee() : double
+ setTotalFee(fee : double) : void
+ isActive() : boolean
+ setActive(active : boolean) : void
+ getReservedMinutes() : int
+ setReservedMinutes(minutes : int) : void
+ getEstimatedFee() : double
+ setEstimatedFee(fee : double) : void
+ isPaid() : boolean
+ setPaid(paid : boolean) : void
+ getPaymentMethod() : String
+ setPaymentMethod(method : String) : void
+ closeTicket(exitTime : LocalDateTime) : void
+ calculateDurationMinutes() : long
+ generateFee() : void
+ pay(method : String) : void
+ getTicketIDCode() : String
+ toString() : String
```

ReservationDialog
```
- parentGUI : ClientGUI
- client : Client
- slot : ParkingSlot
- entryTime : LocalDateTime
- durationSpinner : JSpinner

+ ReservationDialog(gui : ClientGUI, client : Client, slot : ParkingSlot)
+ showDialog() : void
+ onConfirm() : void
```

TicketService
```
- parkingSystem : ParkingSystem
- paymentProcessor : PaymentProcessor

+ TicketService(system : ParkingSystem, processor : PaymentProcessor)
+ createTicket(c : Client, v : Vehicle, slot : ParkingSlot) : Ticket
+ closeTicket(ticketId : int, exitTime : LocalDateTime) : void
+ listActiveTickets() : List<Ticket>
+ extendExpiration(ticketId : int, minutes : int) : void
+ runOverstayChecks() : void
+ payTicket(ticketId : int, method : String) : void
+ getTicketFee(ticketId : int) : double
+ generateReceipt(ticketId : int) : String
+ findTicketById(ticketId : int) : Ticket
```

PaymentProcessor
```
+ calculateFee(t : Ticket, hourlyRate : double) : double
+ takePayment(t : Ticket, cardToken : String) : boolean
+ generateReceipt(t : Ticket) : String
```

PaymentGateway
```
+ authorize(amount : double, cardToken :String) : String
+ refund(paymentID : String, amount : double) : boolean
```

Main
```
+ Main()
+ main(args : String[]) : void
```

AdminGUI
```
- frame : JFrame
- client : EventStreamClient
- connected : boolean
- currentUser : User
- adminPanel : AdminPanel

+ AdminGUI()
+ start() : void
+ connect(host : String, port : int) : void
+ login(email : String, password : String) : void
+ showAdminPanel() : void
+ logout() : void
+ handleError(message : String) : void
+ getCurrentUser() : User
```

Client
```
+ registeredVehicles : List<vehicle>
+ activeTickets : List<Ticket>
+ ticketHistory : List<Ticket>

+ registerVehicle(vehicle : Vehicle) : void
+ requestParking(slot : ParkingSlot) : Ticket
+ viewActiveTickets() : void
+ viewTicketHistory() : void
+ getRegisteredVehicles() : List<Vehicle>
+ addRegisteredVehicle(v : Vehicle) : void
+ removeRegisteredVehicle(plate : String) : boolean
+ getActiveTickets() : List<Ticket>
+ getTicketHistory() : List<Ticket>
+ addToTicketHistory(t : Ticket) : void
```

EventStreamClient
```
- host : String
- port : int
- socket : Socket
- in : ObjectInputStream
- out : ObjectOutputStream

+ EventStreamClient()
+ connect(host : String, port : int) : void
+ close() : void
+ send(msg : Message) : void
+ receive() : Message
+ login(email : String, password : String) : Message
+ logout() : void
+ registerAccount(data : Map<String,String>) : Message
+ addSlotsOnServer(slots : List<ParkingSlot>) : Message
+ removeSlot(slotId : int) : Message
+ fetchSlotsFromServer() : List<ParkingSlot>
+ fetchTicketsFromServer() : List<Ticket>
+ reserveSlotOnServer(slotId : int, minutes : int) : Message
```

ClientGUI
```
- frame : JFrame
- cardLayout : CardLayout
- rootPanel : JPanel
- loginPanel : LoginPanel
- registerPanel : RegisterPanel
- slotsPanel : SlotsPanel
- currentUser : User
- slots : List<ParkingSlot>
- selectedSlotId : int

+ ClientGUI()
+ start() : void
+ connect(host : String, port : int) : void
+ disconnect() : void
+ login(email : String, password : String) : void
+ logout() : void
+ showLoginPage() : void
+ showSlotsPage() : void
+ createAccount(firstName : String, lastName : String, email : String, password : String) : void
+ refreshSlots() : void
+ issueTicket(slotId : int) : void
+ closeTicket(ticketId : int) : void
+ pay(ticketId : int, method : String) : void
+ onSpaceUpdate(msg : Message) : void
+ updateUI() : void
+ handleError(message : String) : void
```

AdminPanel
```
- parentGUI : AdminGUI
- slotsPanel : SlotsPanel
- dashboardTable : JTable
- ticketsTable : JTable

+ AdminPanel(parent : AdminGUI)
+ logout() : void
+ buildDashboard() : void
+ refreshDashboard() : void
+ refreshReports() : void
+ refreshAll() : void
+ getSlotsPanel() : SlotsPanel
```

SlotsPanel
```
gui : ClientGUI
contentPanel : JPanel
contentLayout : CardLayout
accountInfoArea : JTextArea
gateAttendantBtn : JButton
currentFloor : int
currentSection : String
breadcrumbLabel : JLabel
CARD_SLOTS : String
CARD_VEHICLEREG : String
CARD_ACCOUNT : String
CARD_RESERVATIONS : String
CARD_ADMIN_SLOTS : String
CARD_ADMIN_PANEL : String
viewSlotsBtn : JButton
vehicleRegOrAddSlotsBtn : JButton
viewAccountBtn : JButton
viewReservations : JButton
logoutBtn : JButton
gridPanel : JPanel
filterCombo : JComboBox<Object>
currentSlots : List<ParkingSlot>
reservationsModel : DefaultListModel<String>
reservationsList : JList<String>
myReservations : List<Ticket>
ticketsTable : JTable

+ openAdminSlotConfigDialog(ParkingSlot): void
+ buildAccountInfoText() : String
+ getSelectedFilterType() : VehicleType
+ updateBreadcrumb() : void
+ refreshGrid() : void
+ loadSlots(List<ParkingSlot>) : void
+ buildAdminSlotsCard() : JPanel
+ paySelectedReservation() : void
+ SlotsPanel(ClientGUI)
+ updateRoleUI() : void
+ isAdmin() : boolean
+ refreshAccountInfo() : void
+ buildSlotsCard() : JPanel
+ buildRegisterVehicleCard() : JPanel
+ buildAccountCard() : JPanel
+ buildReservationCard() : JPanel
+ addReservation(Ticket) : void
+ openReservationDialog(ParkingSlot) : void
+ refreshReservationsList() : void
```

ParkingSlot
```
- slotId : int
- occupied : boolean
- vehicle : Vehicle
- allowedType : VehicleType
- hourlyRate : double
- outOfService : boolean
- floor : int
- section : String

+ ParkingSlot(slotId : int, allowedType : VehicleType, hourlyRate : double)
+ ParkingSlot(floor : int, section : String, slotId : int)
+ getSlotId() : int
+ setSlotId(id : int) : void
+ isOccupied() : boolean
+ setOccupied(occupied : boolean) : void
+ getVehicle() : Vehicle
+ setVehicle(v : Vehicle) : void
+ getAllowedType() : VehicleType
+ setAllowedType(t : VehicleType) : void
+ getHourlyRate() : double
+ setHourlyRate(rate : double) : void
+ isOutOfService() : boolean
+ setOutOfService(flag : boolean) : void
+ getFloor() : int
+ setFloor(floor : int) : void
+ getSection() : String
+ setSection(section : String) : void
+ assignVehicle(v : Vehicle) : void
+ removeVehicle() : void
+ getCompositeId() : String
```

<<Abstract>>
Vehicle
```
+ plateNumber : String
+ brand : String
+ model : String
+ color : String
+ type : VehicleType

+ calculateFee(duration : int) : double
+ getPlateNumber() : String
+ setPlateNumber(p : String) : void
+ getBrand() : String
+ setBrand(b : String) : void
+ getModel() : String
+ setModel(m : String) : void
+ getColor() : String
+ setColor(c : String) : void
+ getType() : VehicleType
+ setType(type : VehicleType) : void
```

<<Enum>>
VehicleType
```
CAR
MOTORCYCLE
EV
TRUCK
VAN
BUS
SCOOTER
```

<<Enum>>
VehicleType
```
BLACK
WHITE
BLUE
RED
SILVER
GREY
OTHER
```

SlotCell.java
```
gui : ClientGUI
slot : ParkingSlot
filterType : VehicleType
hovered : boolean

+ SlotCell(ClientGUI, ParkingSlot, VehicleType) updateColors()
+ isSlotOwnedByCurrentUser(ParkingSlot)
+ isSlotCompatibleWithFilter(ParkingSlot, VehicleType)
```

Bike
```
+ Bike(...)
+ calculateFee(durationMinutes : long) : double
```

Scooter
```
+ Scooter(...)
+ calculateFee(durationMinutes : long) : double
```

Compact
```
+ Compact(...)
+ calculateFee(durationMinutes : long) : double
```

Van
```
+ Van(...)
+ calculateFee(durationMinutes : long) : double
```

Truck
```
+ Truck(...)
+ calculateFee(durationMinutes : long) : double
```

Bus
```
+ Bus(...)
+ calculateFee(durationMinutes : long) : double
```

EV
```
+ EV(...)
+ calculateFee(durationMinutes : long) : double
```

Car
```
+ Car(...)
+ calculateFee(durationMinutes : long) : double
```

SUV
```
+ SUV(...)
+ calculateFee(durationMinutes : long) : double
```

LoginPanel
```
- parentGUI : ClientGUI
- emailField : JTextField
- passwordField : JPasswordField
- role : String

+ LoginPanel(gui : ClientGUI)
+ setRole(role : String) : void
+ getEmail() : String
+ getPassword() : String
+ onLoginClicked() : void
+ onCreateAccountClicked() : void
```

RegisterPanel.java
```
- parentGUI : ClientGUI
- firstNameField : JTextField
- lastNameField : JTextField
- emailField : JTextField
- passwordField : JPasswordField

+ RegisterPanel(gui : ClientGUI)
+ getFirstName() : String
+ getLastName() : String
+ getEmail() : String
+ getPassword() : String
+ onRegisterClicked() : void
+ onBackToLoginClicked() : void
```