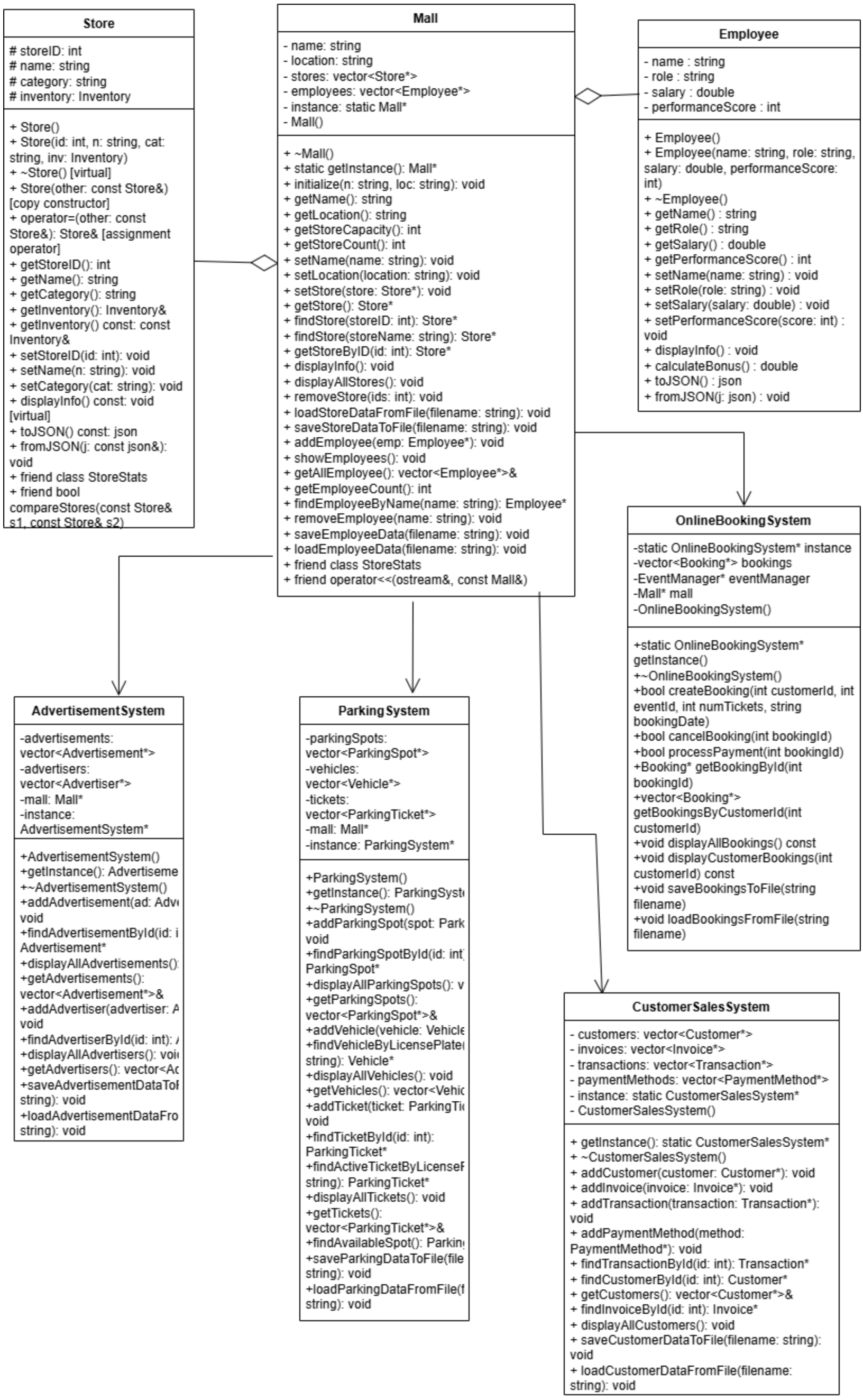
Store
storeID: int
name: string
category: string
inventory: Inventory
+ Store()
+ Store(id: int, n: string, cat: string, inv: Inventory)
+ ~Store() [virtual]
+ Store(other: const Store&) [copy constructor]
+ operator=(other: const Store&): Store& [assignment operator]
+ getStoreID(): int
+ getName(): string
+ getCategory(): string
+ getInventory(): Inventory&
+ getInventory() const: const Inventory&
+ setStoreID(id: int): void
+ setName(n: string): void
+ setCategory(cat: string): void
+ displayInfo() const: void [virtual]
+ toJSON() const: json
+ fromJSON(j: const json&): void
+ friend class StoreStats
+ friend bool compareStores(const Store& s1, const Store& s2)
Mall
- name: string
- location: string
- stores: vector<Store*>
- employees: vector<Employee*>
- instance: static Mall*
- Mall()
+ ~Mall()
+ static getInstance(): Mall*
+ initialize(n: string, loc: string): void
+ getName(): string
+ getLocation(): string
+ getStoreCapacity(): int
+ getStoreCount(): int
+ setName(name: string): void
+ setLocation(location: string): void
+ setStore(store: Store*): void
+ getStore(): Store*
+ findStore(storeID: int): Store*
+ findStore(storeName: string): Store*
+ getStoreByID(id: int): Store*
+ displayInfo(): void
+ displayAllStores(): void
+ removeStore(ids: int): void
+ loadStoreDataFromFile(filename: string): void
+ saveStoreDataToFile(filename: string): void
+ addEmployee(emp: Employee*): void
+ showEmployees(): void
+ getAllEmployee(): vector<Employee*>&
+ getEmployeeCount(): int
+ findEmployeeByName(name: string): Employee*
+ removeEmployee(name: string): void
+ saveEmployeeData(filename: string): void
+ loadEmployeeData(filename: string): void
+ friend class StoreStats
+ friend operator<<(ostream&, const Mall&)
Employee
- name : string
- role : string
- salary : double
- performanceScore : int
+ Employee()
+ Employee(name: string, role: string, salary: double, performanceScore: int)
+ ~Employee()
+ getName() : string
+ getRole() : string
+ getSalary() : double
+ getPerformanceScore() : int
+ setName(name: string) : void
+ setRole(role: string) : void
+ setSalary(salary: double) : void
+ setPerformanceScore(score: int) : void
+ displayInfo() : void
+ calculateBonus() : double
+ toJSON() : json
+ fromJSON(j: json) : void
OnlineBookingSystem
-static OnlineBookingSystem* instance
-vector<Booking*> bookings
-EventManager* eventManager
-Mall* mall
-OnlineBookingSystem()
+static OnlineBookingSystem* getInstance()
+~OnlineBookingSystem()
+bool createBooking(int customerId, int eventId, int numTickets, string bookingDate)
+bool cancelBooking(int bookingId)
+bool processPayment(int bookingId)
+Booking* getBookingById(int bookingId)
+vector<Booking*> getBookingsByCustomerId(int customerId)
+void displayAllBookings() const
+void displayCustomerBookings(int customerId) const
+void saveBookingsToFile(string filename)
+void loadBookingsFromFile(string filename)
AdvertisementSystem
-advertisements: vector<Advertisement*>
-advertisers: vector<Advertiser*>
-mall: Mall*
-instance: AdvertisementSystem*
+AdvertisementSystem()
+getInstance(): Advertiseme
+~AdvertisementSystem()
+addAdvertisement(ad: Adv
void
+findAdvertisementById(id: i
Advertisement*
+displayAllAdvertisements():
+getAdvertisements(): vector<Advertisement*>&
+addAdvertiser(advertiser: A
void
+findAdvertiserById(id: int): A
+displayAllAdvertisers(): voi
+getAdvertisers(): vector<Ac
+saveAdvertisementDataTof
string): void
+loadAdvertisementDataFro
string): void
ParkingSystem
-parkingSpots: vector<ParkingSpot*>
-vehicles: vector<Vehicle*>
-tickets: vector<ParkingTicket*>
-mall: Mall*
-instance: ParkingSystem*
+ParkingSystem()
+getInstance(): ParkingSyst
+~ParkingSystem()
+addParkingSpot(spot: Park
void
+findParkingSpotById(id: int
ParkingSpot*
+displayAllParkingSpots(): v
+getParkingSpots(): vector<ParkingSpot*>&
+addVehicle(vehicle: Vehicle
+findVehicleByLicensePlate(
string): Vehicle*
+displayAllVehicles(): void
+getVehicles(): vector<Vehic
+addTicket(ticket: ParkingTi
void
+findTicketById(id: int): ParkingTicket*
+findActiveTicketByLicenseF
string): ParkingTicket*
+displayAllTickets(): void
+getTickets(): vector<ParkingTicket*>&
+findAvailableSpot(): Parking
+saveParkingDataToFile(file
string): void
+loadParkingDataFromFile(f
string): void
CustomerSalesSystem
- customers: vector<Customer*>
- invoices: vector<Invoice*>
- transactions: vector<Transaction*>
- paymentMethods: vector<PaymentMethod*>
- instance: static CustomerSalesSystem*
- CustomerSalesSystem()
+ getInstance(): static CustomerSalesSystem*
+ ~CustomerSalesSystem()
+ addCustomer(customer: Customer*): void
+ addInvoice(invoice: Invoice*): void
+ addTransaction(transaction: Transaction*): void
+ addPaymentMethod(method: PaymentMethod*): void
+ findTransactionById(id: int): Transaction*
+ findCustomerById(id: int): Customer*
+ getCustomers(): vector<Customer*>&
+ findInvoiceById(id: int): Invoice*
+ displayAllCustomers(): void
+ saveCustomerDataToFile(filename: string): void
+ loadCustomerDataFromFile(filename: string): void