UML Diagrams

Interfaces (top)

<<Interface>> IRentalMgt
+createRental(customerId, playstationId, hours) : boolean
+endRental(rentalId) : boolean
+getAllRentals() : List<RentalTransaction>
+getRentalsByDate(date) : List<RentalTransaction>
+getAvailablePlayStations() : List<PlayStation>
+getPlayStationById(id) : PlayStation
+getActiveRentals() : List<RentalTransaction>

<<Interface>> IFoodBeverageMgt
+createFoodBeverageTransaction(customerId, items) : boolean
+getAllFBTransactions() : List<FoodBeverageTransaction>
+getFBTransactionsByDate(date) : List<FoodBeverageTransaction>
+getAllItems() : List<FoodBeverage>
+getItemById(id) : FoodBeverage

<<Interface>> IFinancialDashboard
+getTotalRevenue(startDate, endDate) : double
+getTotalRentalRevenue(startDate, endDate) : double
+getTotalFBRevenue(startDate, endDate) : double
+getProfit(startDate, endDate) : double
+getDailyRevenue(startDate, endDate) : Map<LocalDate,Double>

<<Interface>> IExpenseReport
+getAllExpenses() : List<Expense>
+getExpensesByDate(startDate, endDate) : List<Expense>
+getExpensesByCategory(category) : List<Expense>
+getTotalExpenseAmount(startDate, endDate) : double

<<Interface>> ICustomerMgt
+addCustomer(name, phone) : int
+getCustomerById(id) : Customer

<<Interface>> IExpenseMgt
+addExpense(description, amount, date) : boolean
+addExpense(description, amount, date, category) : boolean
+updateExpense(expenseId, description, amount, date) : boolean
+deleteExpense(expenseId) : boolean

Classes

Main
-system PlaystationRentalSystem
-scanner Scanner
-dateFormatter DateTimeFormatter
+main(args) : void
-showCashierMenu() : void
-showOwnerMenu() : void
-createRentalTransaction() : void
-endRentalTransaction() : void
-createFBTransaction() : void
-showFinancialDashboard() : void
-addExpense() : void
-viewExpenses() : void
-getIntInput() : int
-getDateInput() : LocalDate

PlayStationRentalSystem
-rentalManager RentalManager
-fbManager FoodBeverageManager
-expenseManager ExpenseManager
-financialManager FinancialManager
-customerManager CustomerManager
+PlayStationRentalSystem()
-connectComponents() : void
-initializeSystem() : void
+addCustomer(name, phone) : int
+getCustomerById(id) : Customer
+getActiveRentals() : List<RentalTransaction>
+createRental(customerId, playstationId, hours) : boolean
+endRental(rentalId) : boolean
+getAvailablePlayStations() : List<PlayStation>
+getPlayStationById(playstationId) : PlayStation
+createFBTransaction(customerId, items) : boolean
+getAllFBItems() : List<FoodBeverage>
+getTotalRevenue(startDate, endDate) : double
+getProfit(startDate, endDate) : double
+getDailyRevenue(startDate, endDate) : Map<LocalDate,Double>
+addExpense(description, amount, date, category) : boolean
+getAllExpenses() : List<Expense>
+getExpensesByDate(startDate, endDate) : List<Expense>
+getTotalExpenses(startDate, endDate) : double

CustomerManager
-customers List<Customer>
-nextCustomerId int
+CustomerManager()
+addCustomer(name, phone) : int
+getCustomerById(id) : Customer

FoodBeverageManager
-transactions List<FoodBeverageTransaction>
-inventory List<FoodBeverage>
-nextTransactionId int
+FoodBeverageManager()
+addFoodBeverage(item) : void
+getItemById(id) : FoodBeverage
+getAllItems() : List<FoodBeverage>
+createFoodBeverageTransaction(customerId, items) : boolean
+getAllFBTransactions() : List<FoodBeverageTransaction>
+getFBTransactionsByDate(date) : List<FoodBeverageTransaction>

FinancialManager
-rentalManager IRentalMgt
-fbManager IFoodBeverageMgt
-expenseManager IExpenseReport
+FinancialManager()
+connectTo(rentalManager) : void
+connectTo(fbManager) : void
+connectTo(expenseManager) : void
+getTotalRevenue(startDate, endDate) : double
+getTotalRentalRevenue(startDate, endDate) : double
+getTotalFBRevenue(startDate, endDate) : double
+getTotalExpenses(startDate, endDate) : double
+getProfit(startDate, endDate) : double
+getDailyRevenue(startDate, endDate) : Map<LocalDate,Double>

RentalManager
-rentals List<RentalTransaction>
-playstations List<PlayStation>
-nextRentalId int
+addPlayStation(ps) : void
+getAvailablePlayStations() : List<PlayStation>
+getPlayStationById(id) : PlayStation
+createRental(customerId, playstationId, hours) : boolean
+endRental(rentalId) : boolean
+getActiveRentals() : List<RentalTransaction>
+getAllRentals() : List<RentalTransaction>
+getRentalsByDate(date) : List<RentalTransaction>

ExpenseManager
-expenses List<Expense>
-nextExpenseId int
+ExpenseManager()
+addExpense(description, amount, date) : boolen
+addExpense(description, amount, date, category) : boolean
+updateExpense(expenseId, description, amount, date) : boolean
+deleteExpense(expenseId) : boolean
+getAllExpenses() : List<Expense>
+getExpensesByDate(startDate, endDate) : List<Expense>
+getExpensesByCategory(category) : List<Expense>
+getTotalExpenseAmount(startDate, endDate) : double

<<Interface>> ICustomerMgt
+addCustomer(name, phone) : int
+getCustomerById(id) : Customer

Customer
-id int
-name String
-phone String
+getId() : int
+getName() : String
+setName(name) : void
+getPhone() : String
+setPhone(phone) : void
+toString() : String

<<Interface>> IFoodBeverageMgt
+createFoodBeverageTransaction(customerId, items) : boolean
+getAllFBTransactions() : List<FoodBeverageTransaction>
+getFBTransactionsByDate(date): List<FoodBeverageTransaction>
+getAllItems() : List<FoodBeverage>
+getItemById(id) : FoodBeverage

<<Interface>> IRentalMgt
+createRental(customerId, playstationId, hours) : boolean
+endRental(rentalId) : boolean
+getAllRentals() : List<RentalTransaction>
+getRentalsByDate(date) : List<RentalTransaction>
+getAvailablePlayStations() : List<PlayStation>
+getPlaystationById(id) : PlayStation
+getActiveRentals() : List<RentalTransaction>

<<Interface>> IExpenseReport
+getAllExpenses() : List<Expense>
+getExpensesByDate(startDate, endDate) : List<Expense>
+getExpensesByCategory(category) : List<Expense>
+getTotalExpenseAmount(startDate, endDate) : double

<<Interface>> IExpenseMgt
+addExpense(description, amount, date) : boolean
+addExpense(description, amount, date, category) : boolean
+updateExpense(expenseId, description, amount, date) : boolean
+deleteExpense(expenseId) : boolean

FoodBeverageTransaction
-id int
-customerId int
-items Map<Integer,Integer>
-timestamps LocalDateTime
-totalAmount double
+getId() : int
+getCustomerId() : int
+getItems() : Map<Integer,Integer>
+getTimestamp() : LocalDateTime
+getTotalAmount() : double
+getTransactionDate() : LocalDate
+toString() : String

FoodBeverage
-id int
-name String
-price double
-stock int
-category String
+FoodBeverage(id, name, price, stock, category)
+getId() : int
+getName() : String
+getPrice() : double
+setPrice(price) : void
+getStock() : int
+setStock(stock) : void
+getCategory() : String
+decreaseStock(quantity) : boolean
+toString() : String

<<Interface>> IFinancialDashboard
+getTotalRevenue(startDate, endDate) : double
+getTotalRentalRevenue(startDate, endDate) : double
+getTotalFBRevenue(startDate, endDate) : double
+getTotalExpenses(startDate, endDate) : double
+getProfit(startDate, endDate) : double
+getDailyRevenue(startDate, endDate) : Map<LocalDate,Double>

PlayStation
-id int
-model String
-isAvailable boolean
-hourlyRate double
+PlayStation(id, model, hourlyRate)
+getId() : int
+getModel() : String
+isAvailable() : boolean
+setAvailable(available) : void
+getHourlyRate() : double
+setHourlyRate(hourlyRate) : void
+toString() : String

RentalTransaction
-id int
-customerId int
-playstationId int
-startTime LocalDateTime
-endTime LocalDateTime
-hours int
-amount double
-active boolean
+RentalTransaction(id, customerId, playstationId, startTime, hours, amount, active)
+getId() : int
+getCustomerId() : int
+getPlaystationId() : int
+getStartTime() : LocalDateTime
+getEndTime() : LocalDateTime
+getHours() : int
+getAmount() : double
+isActive() : boolean
+endRental() : void
+getTransactionDate() : LocalDate

Expense
-id int
-description String
-amount double
-date LocalDate
-category String
+Expense(id, description, amount, date, category)
+getId() : int
+getDescription() : String
+setDescription(description) : void
+getAmount() : double
+setAmount(amount) : void
+getDate() : LocalDate
+setDate(date) : void
+getCategory() : String
+setCategory(category) : void
+toString() : String

Component Diagram

- «component» :CustomerManager — ICustomerMgt
- «component» :ExpenseManager — IExpenseMgt, IExpenseReport
- «component» :FinancialManager — IFinancialDashboard, IFoodBeverageMgt — «component» :FoodBeverageManager
- «component» :RentalManager — IRentalMgt

Use Case Diagram

Actors: Cashier, Owner

Use cases:
- Create Rental Transaction
- End Rental Transaction
- Create Food/Beverage Transaction
- View Financial Dashboard
- Record Expense
- View Record Expense