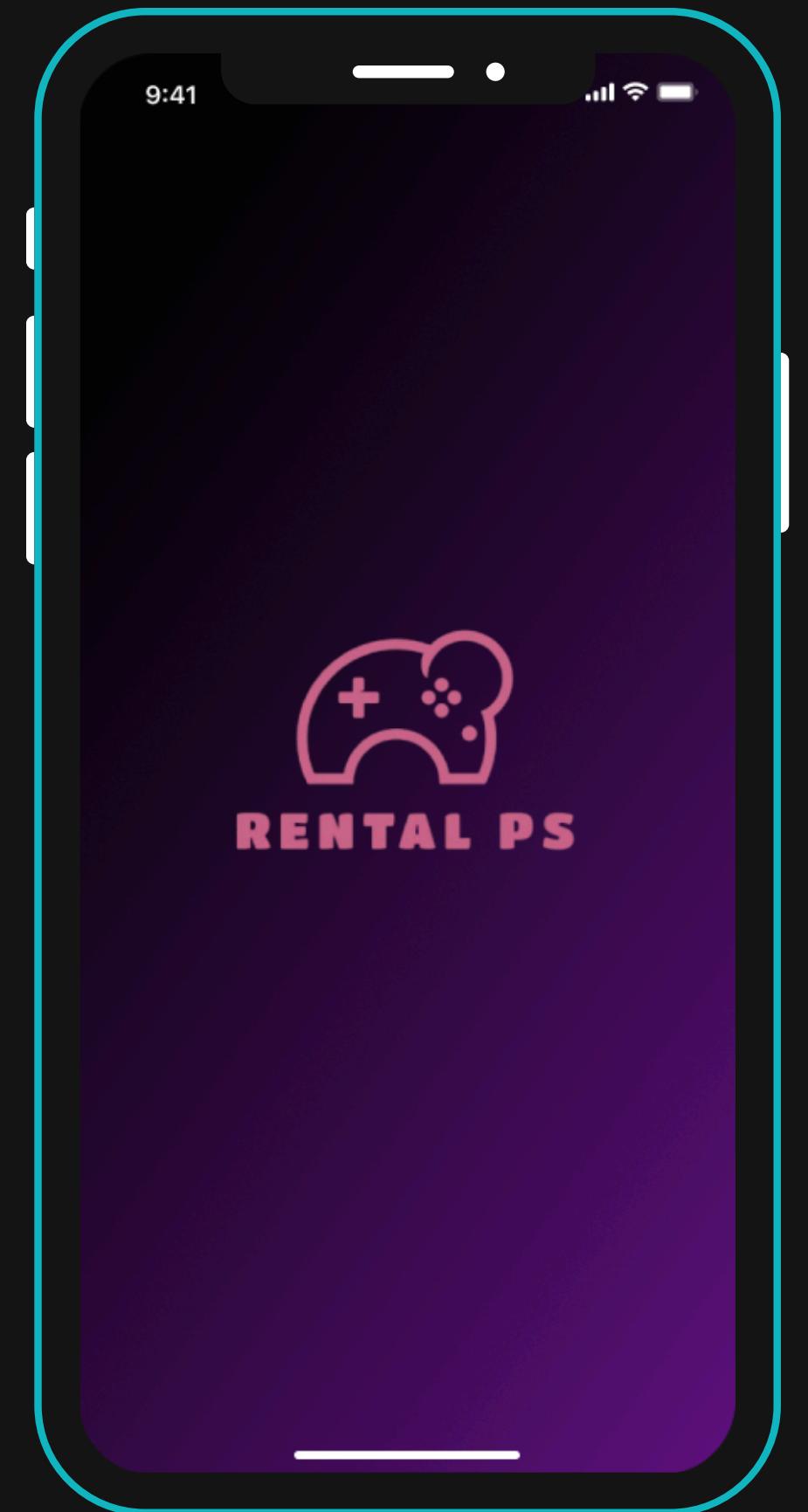


UTS Perangkat
Lunak Berbasis
Komponen

Playstation Rental System

by
Muhammad Bintang Indra Hidayat (2208107010023)
Shofia Nurul Huda (2208107010015)



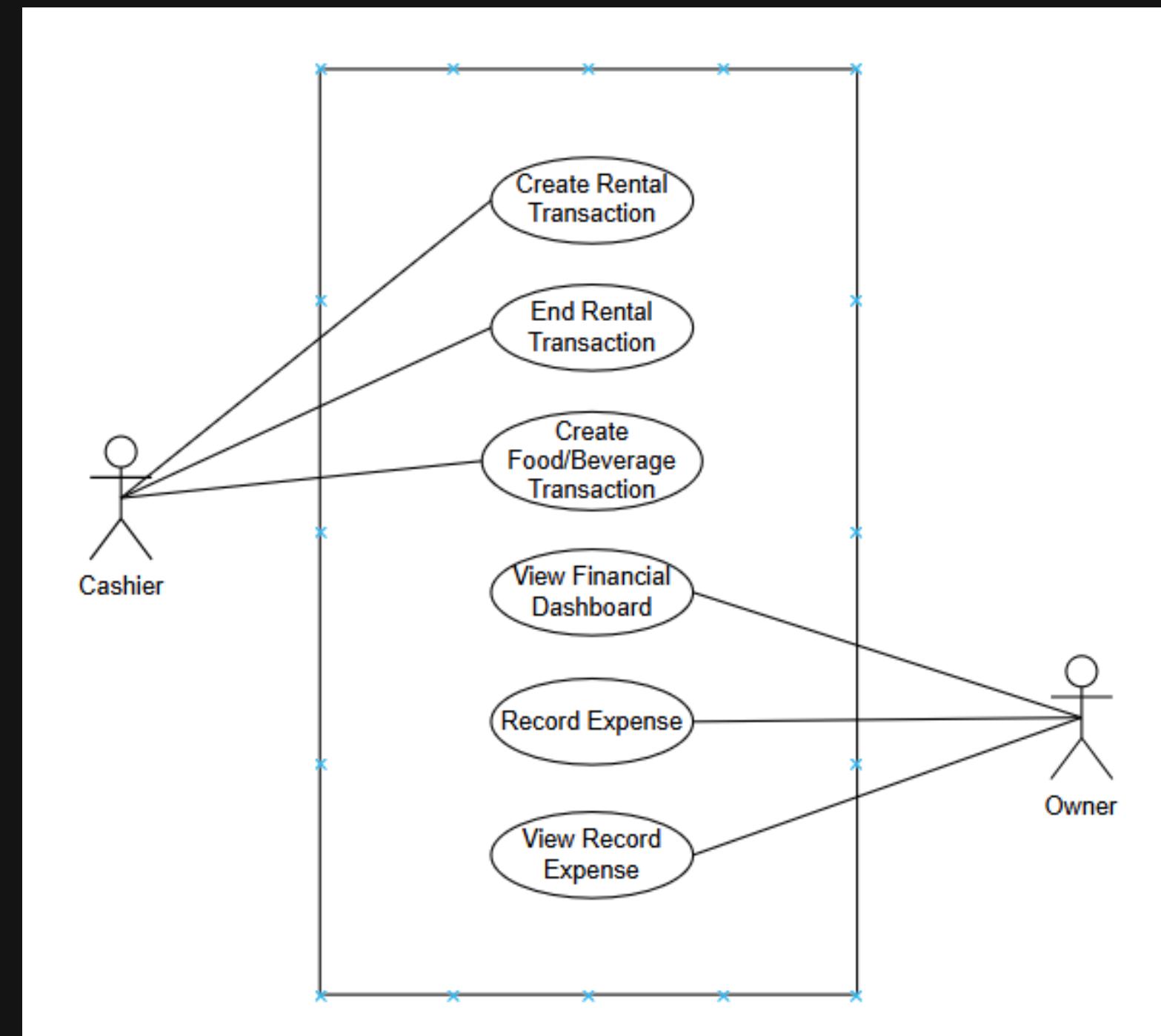


Apa itu PlayStation Rental System?

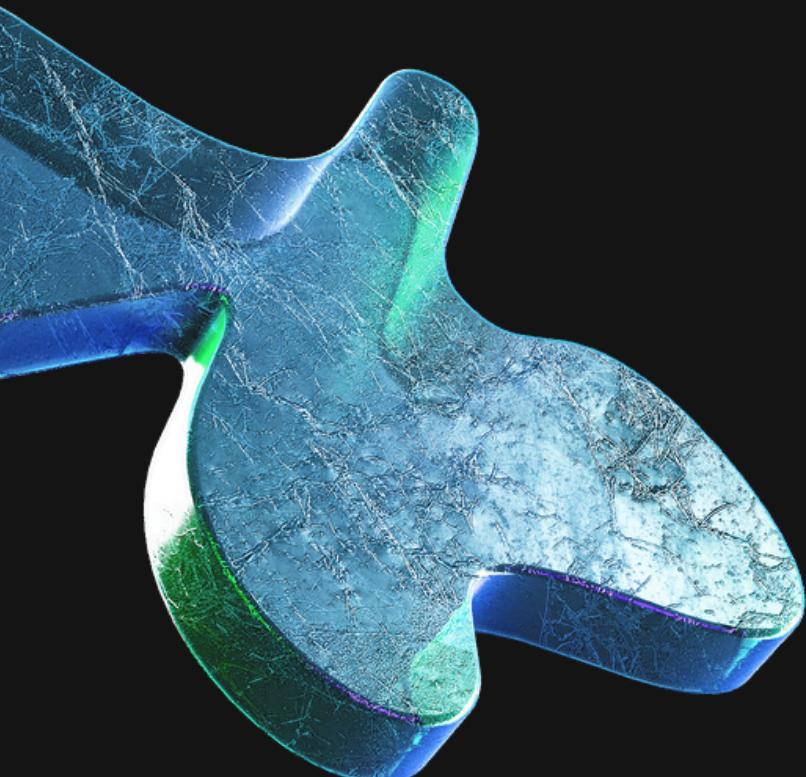


PlayStation Rental System adalah sistem yang mengelola penyewaan PlayStation, transaksi makanan/minuman, serta pencatatan keuangan. Cashier menangani transaksi rental dan penjualan, sementara Owner memantau keuangan dan mencatat pengeluaran, memastikan operasional berjalan efisien.

Use Case Diagram



Main Success Scenario



Create Rental Playstation

Name: Create Rental Transaction
Initiator: Cashier
Goal: Record a new PlayStation rental transaction in the system
Main Success Scenario: <ol style="list-style-type: none">1. Cashier selects "Mencatat Transaksi Sewa PS" from the cashier menu2. System displays all available PlayStation units with their IDs, models, and hourly rates3. Cashier enters customer ID (1 for walk-in customers or existing customer ID)4. Cashier selects PlayStation ID from the displayed list5. Cashier enters rental duration in hours6. System calculates total amount based on hourly rate and duration7. System creates rental transaction and marks the PlayStation as unavailable8. System confirms transaction creation with success message9. System returns to cashier menu
Extensions: <ul style="list-style-type: none">1a. No PlayStation units are available:<ul style="list-style-type: none">- System displays "Tidak ada PlayStation yang tersedia saat ini"- System returns to cashier menu4a. Selected PlayStation ID is invalid or unavailable:<ul style="list-style-type: none">- System displays "PlayStation tidak tersedia atau tidak ditemukan"- Transaction creation is aborted- System returns to cashier menu5a. Entered hours value is not a valid number:<ul style="list-style-type: none">- System prompts "Input tidak valid. Masukkan angka"- Cashier re-enters hours value

End Rental Playstation

Name: End Rental Transaction
Initiator: Cashier
Goal: Complete an active PlayStation rental and make the unit available again
Main Success Scenario: <ol style="list-style-type: none">1. Cashier selects "Mengakhiri Sewa PS" from the cashier menu2. System displays list of active rental transactions with their IDs, PlayStation models, customer IDs, and duration3. Cashier enters rental ID to end4. System marks the rental as inactive with current time as end time5. System updates the PlayStation status to available6. System confirms successful transaction completion7. System returns to cashier menu
Extensions: <ul style="list-style-type: none">2a. No active rental transactions exist:<ul style="list-style-type: none">- System displays "Tidak ada transaksi sewa yang sedang berlangsung"- System returns to cashier menu3a. Entered rental ID is invalid or already completed:<ul style="list-style-type: none">- System displays "Rental tidak ditemukan atau sudah selesai"- System returns to cashier menu3b. Entered value is not a valid number:<ul style="list-style-type: none">- System prompts "Input tidak valid. Masukkan angka"- Cashier re-enters rental ID

Create Food/Beverage Transaction

Name: Create Food/Beverage Transaction (Mencatat Transaksi Pembelian Makanan/Minuman)
Initiator: Cashier
Goal: Record a food and beverage purchase transaction
Main Success Scenario: <ol style="list-style-type: none">1. Cashier selects "Mencatat Transaksi Pembelian Makanan/Minuman" from the cashier menu2. System displays available food and beverage items with IDs, names, prices, and stock levels3. Cashier enters customer ID (1 for walk-in customers)4. Cashier enters item ID for purchase (or 0 to complete order)5. If item ID is not 0, cashier enters quantity of the selected item6. System adds item to current order7. Steps 4-6 repeat until cashier enters 0 for item ID8. System creates transaction, updates inventory stock levels9. System displays "Transaksi pembelian berhasil dicatat!"10. System returns to cashier menu
Extensions: <ul style="list-style-type: none">2a. No food/beverage items are available:<ul style="list-style-type: none">- System displays "Tidak ada item yang tersedia."- System returns to cashier menu4a-5a. Entered value is not a valid number:<ul style="list-style-type: none">- System prompts "Input tidak valid. Masukkan angka."- Cashier re-enters the value8a. Transaction creation fails (insufficient stock or invalid items):<ul style="list-style-type: none">- System displays "Gagal mencatat transaksi pembelian. Pastikan data valid."- System returns to cashier menu

View Financial Dashboard

Name: View Financial Dashboard (Melihat Dashboard Keuangan)
Initiator: Owner
Goal: View summary financial information for a specific period
Main Success Scenario: <ol style="list-style-type: none">1. Owner selects "Melihat Dashboard Keuangan" from the owner menu2. Owner enters start date in format dd/MM/yyyy3. Owner enters end date in format dd/MM/yyyy4. System calculates and displays:<ul style="list-style-type: none">- Total expenses for the period- Total revenue for the period- Total profit for the period5. System returns to owner menu
Extensions: <ul style="list-style-type: none">2a-3a. Entered date is in incorrect format:<ul style="list-style-type: none">- System prompts "Format tanggal salah. Masukkan kembali (dd/MM/yyyy):"- Owner re-enters date in correct format

Record Expense

Name: Record Expense (Mencatat Pengeluaran)
Initiator: Owner
Goal: Add a new expense record to the system
Main Success Scenario: <ol style="list-style-type: none">Owner selects "Mencatat Pengeluaran" from the owner menuOwner enters expense descriptionOwner enters expense amountOwner enters expense date in format dd/MM/yyyyOwner enters expense categorySystem creates and stores the expense recordSystem displays "Pengeluaran berhasil dicatat!"System returns to owner menu
Extensions: <ul style="list-style-type: none">3a. Entered amount is not a valid number:<ul style="list-style-type: none">- System consumes the input and continues (as per code implementation)4a. Entered date is in incorrect format:<ul style="list-style-type: none">- System prompts "Format tanggal salah. Masukkan kembali (dd/MM/yyyy):"- Owner re-enters date in correct format6a. Expense recording fails:<ul style="list-style-type: none">- System displays "Gagal mencatat pengeluaran."- System returns to owner menu

View Expense

Name: View Expense Records (Melihat Catatan Pengeluaran)
Initiator: Owner
Goal: View expense records for a specific period
Main Success Scenario: <ol style="list-style-type: none">Owner selects "Melihat Catatan Pengeluaran" from the owner menuOwner enters start date in format dd/MM/yyyyOwner enters end date in format dd/MM/yyyySystem displays list of all expenses in the date range with date, description, amount, and categorySystem calculates and displays total expenses for the periodSystem returns to owner menu
Extensions: <ul style="list-style-type: none">2a-3a. Entered date is in incorrect format:<ul style="list-style-type: none">- System prompts "Format tanggal salah. Masukkan kembali (dd/MM/yyyy):"- Owner re-enters date in correct format4a. No expenses exist for the selected period:<ul style="list-style-type: none">- System displays "Tidak ada catatan pengeluaran pada periode ini."- System returns to owner menu

Interfaces

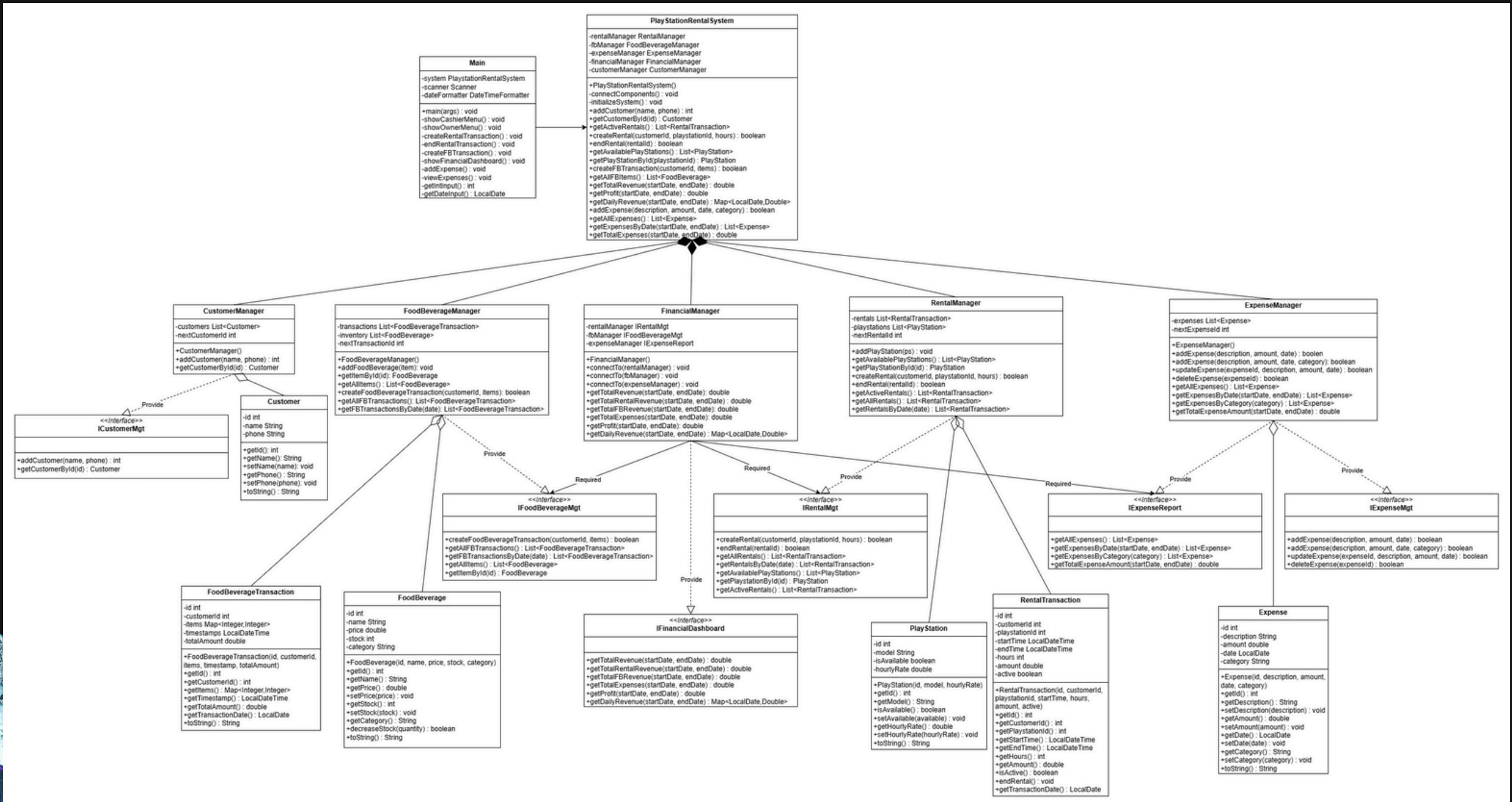
<pre><</interface>> IRentalMgt</pre>	<pre><</interface>> IFoodBeverageMgt</pre>
<pre>+createRental(customerId, playstationId, hours) : boolean +endRental(rentalId) : boolean +getAllRentals() : List<RentalTransaction> +getRentalsByDate(date) : List<RentalTransaction> +getAvailablePlayStations() : List<PlayStation> +getPlaystationById(id) : PlayStation +getActiveRentals() : List<RentalTransaction></pre>	<pre>+createFoodBeverageTransaction(customerId, items) : boolean +getAllFBTransactions() : List<FoodBeverageTransaction> +getFBTransactionsByDate(date) : List<FoodBeverageTransaction> +getAllItems() : List<FoodBeverage> +getItemById(id) : FoodBeverage</pre>
<pre><</interface>> ICustomerMgt</pre>	<pre><</interface>> IExpenseMgt</pre>
<pre>+addCustomer(name, phone) : int +getCustomerById(id) : Customer</pre>	<pre>+addExpense(description, amount, date) : boolean +addExpense(description, amount, date, category) : boolean +updateExpense(expenseId, description, amount, date) : boolean +deleteExpense(expenseId) : boolean</pre>

System Interface

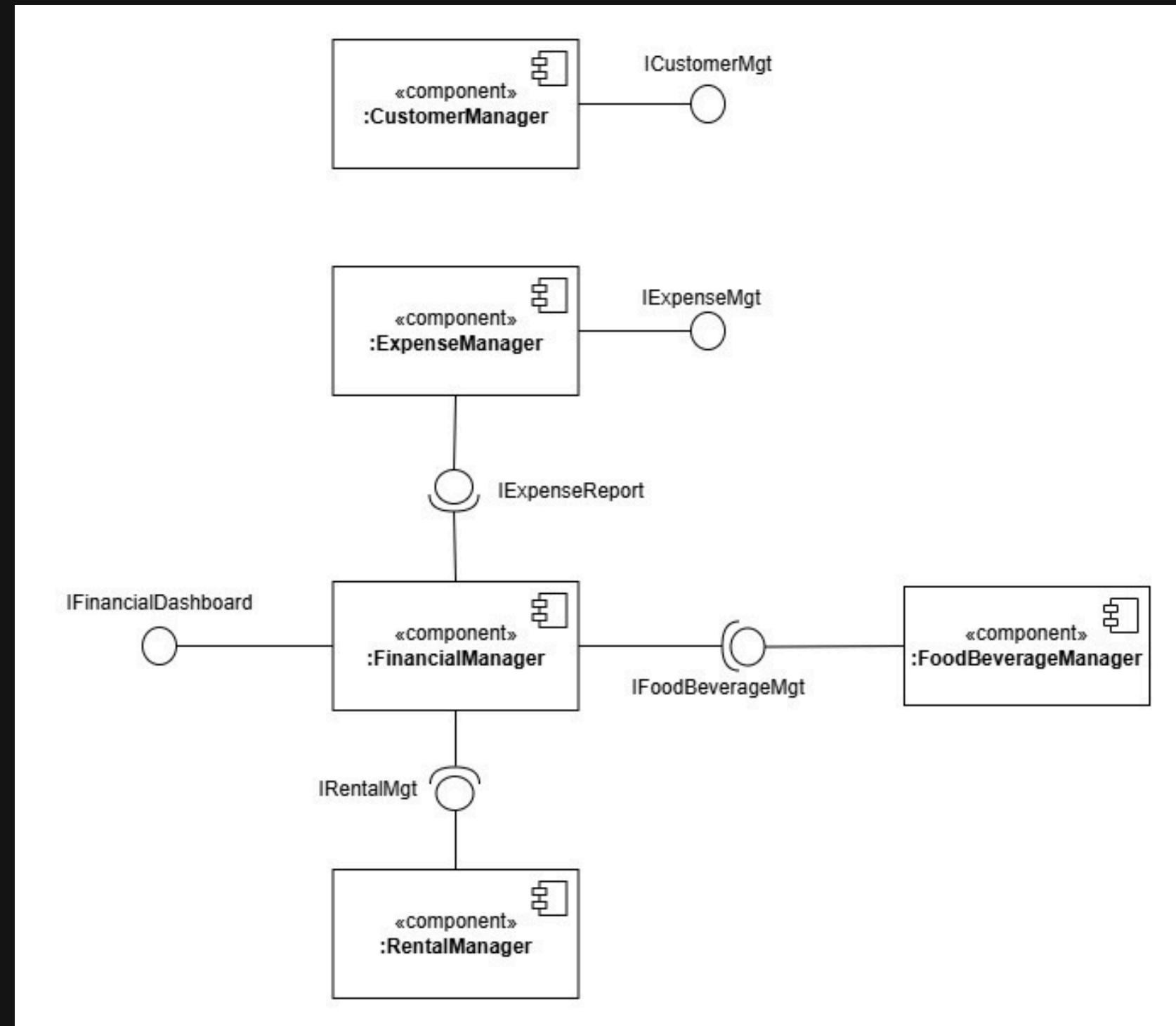
<pre><</interface>> IFinancialDashboard</pre>	<pre><</interface>> IExpenseReport</pre>
<pre>+getTotalRevenue(startDate, endDate) : double +getTotalRentalRevenue(startDate, endDate) : double +getTotalFBReward(startDate, endDate) : double +getTotalExpenses(startDate, endDate) : double +getProfit(startDate, endDate) : double +getDailyRevenue(startDate, endDate) : Map<LocalDate,Double></pre>	<pre>+getAllExpenses() : List<Expense> +getExpensesByDate(startDate, endDate) : List<Expense> +getExpensesByCategory(category) : List<Expense> +getTotalExpenseAmount(startDate, endDate) : double</pre>

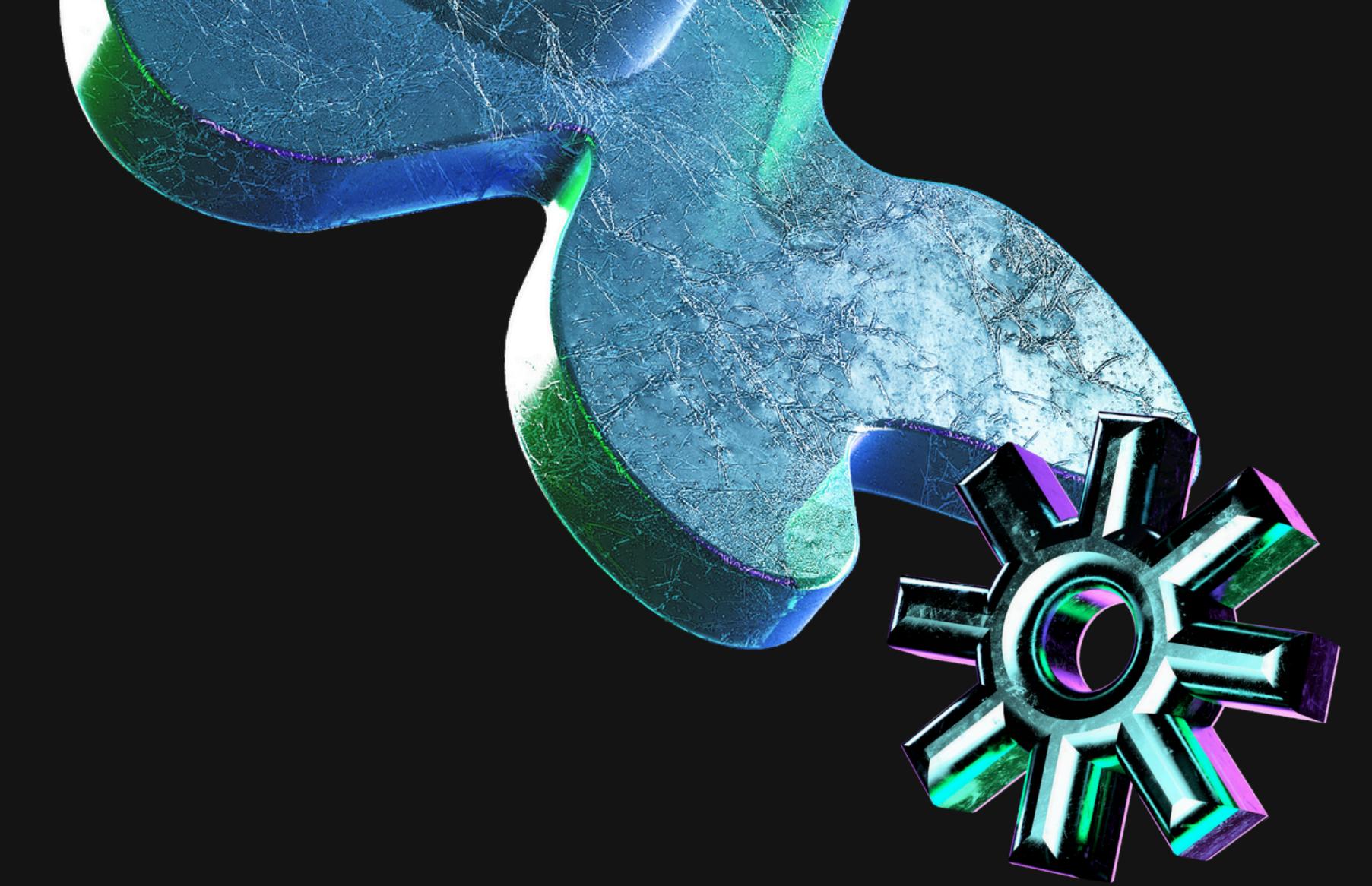
Bussiness Interface

Class Diagram



Component Diagram





Terima Kasih

Semoga Anda mempelajari sesuatu yang baru.