ER Diagram Explanation for Payment Management System

The Entity Relationship Diagram (ERD) represents the core data entities and their relationships within the Payment Management System database. This diagram provides a visual overview of how data is organized, interrelated, and stored to support the application's functionality.

Entities and Their Attributes

1. User

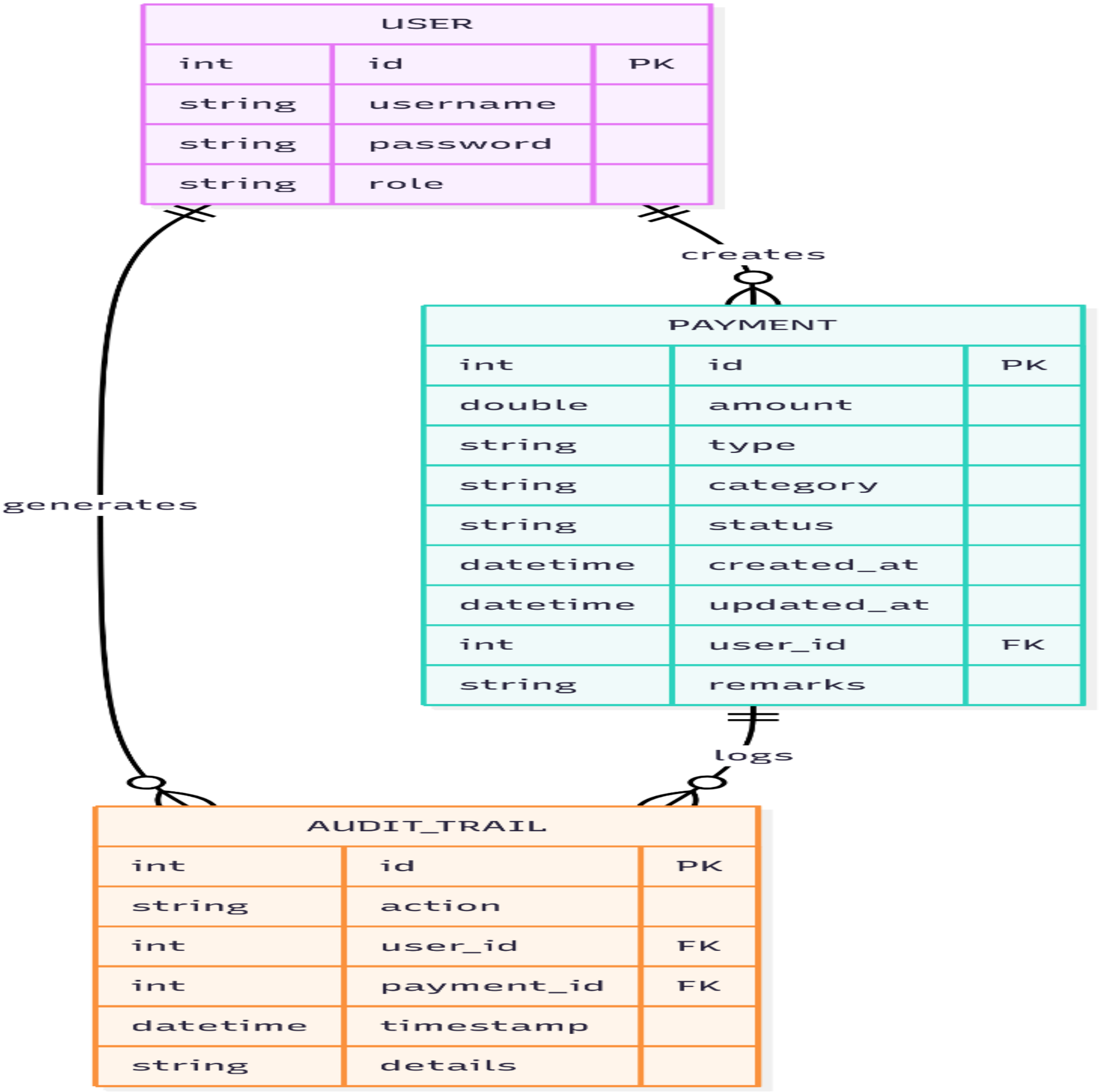
- id (PK): Unique identifier for each user.
- username: Login name of the user.
- password: User authentication credential.
- role: Defines the user's role (e.g., ADMIN, FINANCE_MANAGER, VIEWER), used for access control.

2. Payment

- id (PK): Unique identifier for each payment record.
- amount: Monetary value of the payment.
- type: Indicates whether the payment is incoming or outgoing.
- category: Classification of the payment (e.g., SALARY, VENDOR_PAYMENT, CLIENT_INVOICE).
- status: Current status of payment processing (PENDING, PROCESSING, COMPLETED).
- created_at: Timestamp when the payment was created.
- updated_at: Timestamp of the latest update on the payment.
- user_id (FK): Reference to the user who created or is responsible for the payment.
- remarks: Additional notes regarding the payment.

3. AuditTrail

- id (PK): Unique identifier for audit entries.
- action: Description of the action performed (e.g., Add Payment, Update Status).
- user_id (FK): The user who performed the action.
- payment_id (FK): The payment related to the action.
- timestamp: When the action was performed.
- details: Additional context or information regarding the action.



USER

int	id	PK
string	username	
string	password	
string	role	

PAYMENT

int	id	PK
double	amount	
string	type	
string	category	
string	status	
datetime	created_at	
datetime	updated_at	
int	user_id	FK
string	remarks	

AUDIT_TRAIL

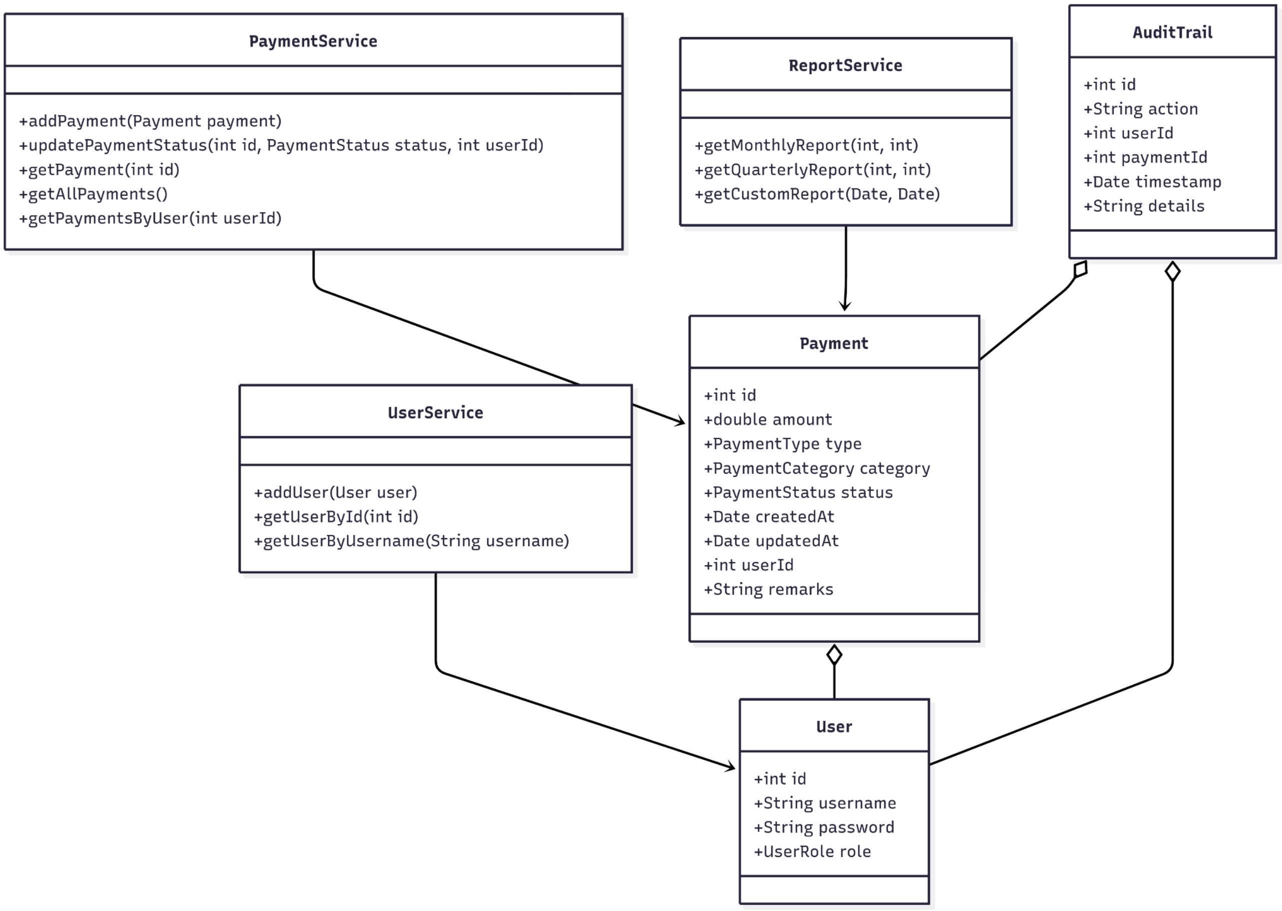
int	id	PK
string	action	
int	user_id	FK
int	payment_id	FK
datetime	timestamp	
string	details	

Class Diagram Explanation

The Class Diagram illustrates key components such as Entities (User, Payment, AuditTrail), Services (UserService, PaymentService, ReportService), and their interactions.

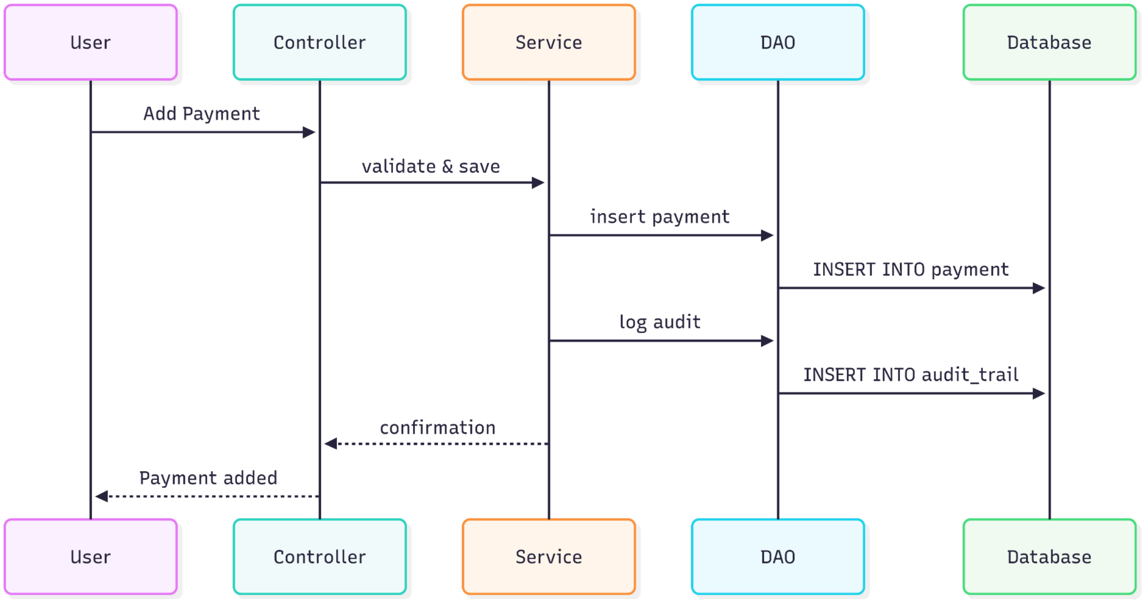
- Entities encapsulate domain data with attributes that map to database fields.
- Services contain business logic, handle validations, and interact with DAOs (Data Access Objects) for persistence.
- Relationships such as the one-to-many between User and Payment, and Payment and AuditTrail, represent data ownership and audit trails.

This layered architecture enforces separation of concerns and facilitates maintainability and extensibility of the payment management system.

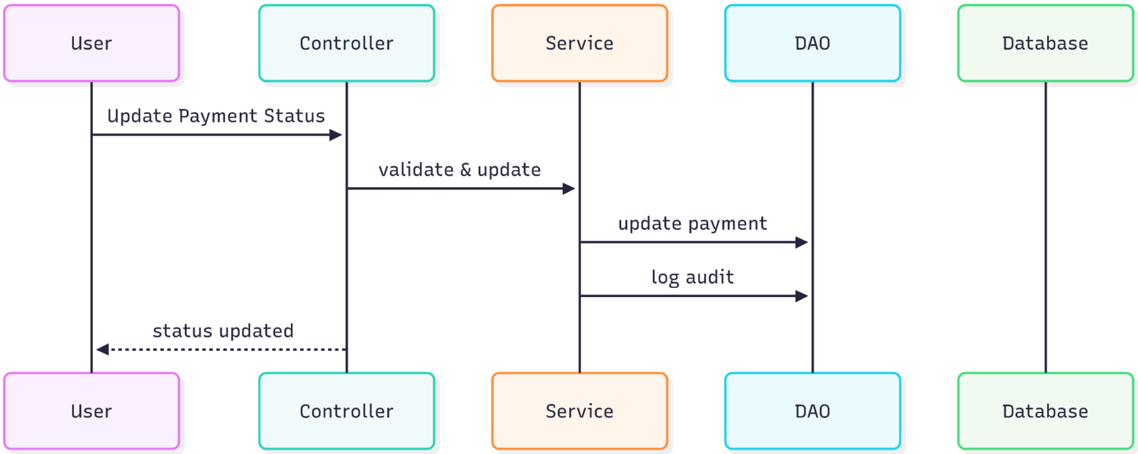


Relationships

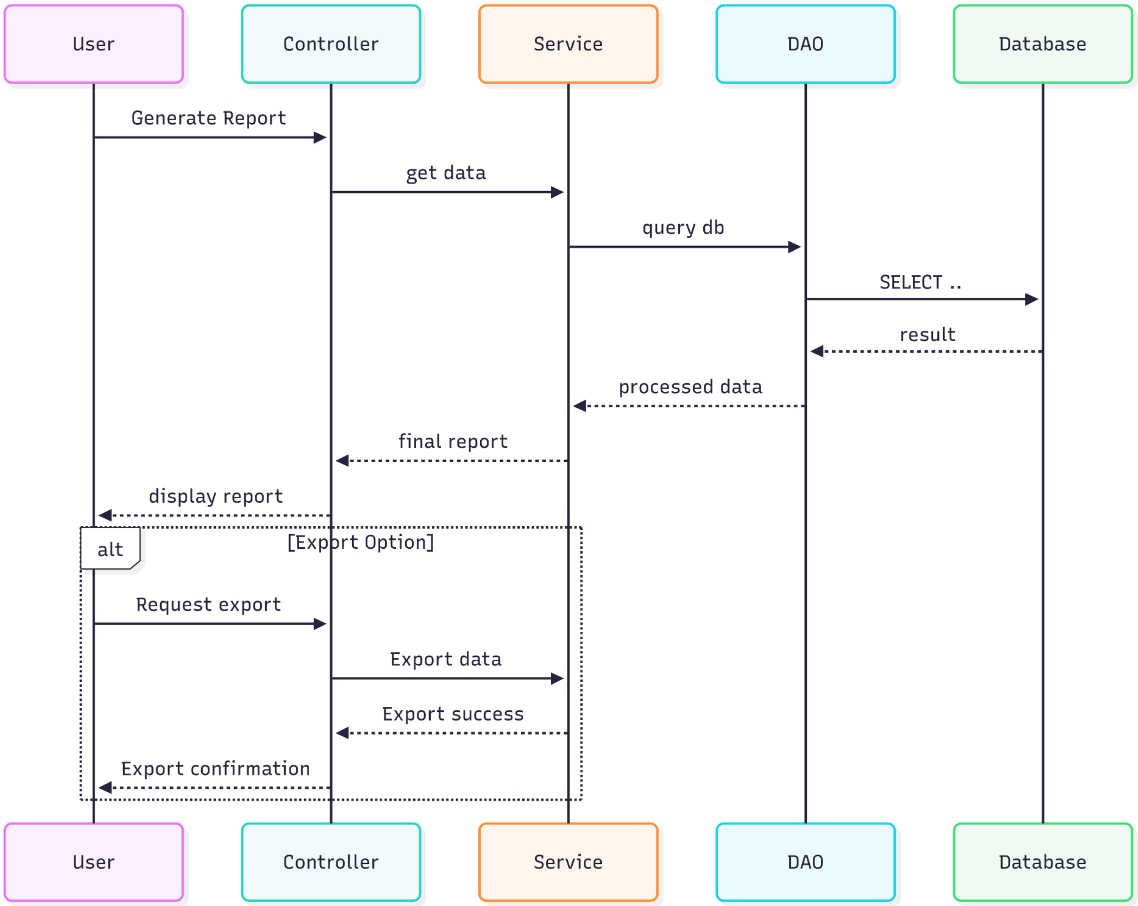
- User to Payment: One-to-Many
  Each user can create multiple payment records. This is represented by a one-to-many relationship where a single user is linked to many payments.



- User to AuditTrail: One-to-Many
  Users perform various actions that are recorded in the audit trail, meaning one user can have many audit entries.



- Payment to AuditTrail: One-to-Many
  Each payment can have multiple audit trail entries capturing all actions performed on it over time.


User
Controller
Service
DAO
Database
Generate Report
get data
query db
SELECT ..
result
processed data
final report
display report
alt
[Export Option]
Request export
Export data
Export success
Export confirmation
User
Controller
Service
DAO
Database