**Netflix System**

The Netflix system is a streaming service that allows users to create an account, log in and out, view and watch content, rate and review content, and manage their subscription. This report includes four basic models of the system: User, Content, Subscription, and Transaction.

Model 1: User

public class User {  
  
 @NotEmpty(message = "ID cannot be empty")  
 @Size(min = 3, max = 20, message = "ID must be between 3 and 20 characters")  
 private String id;  
  
 @NotEmpty(message = "Name is required")  
 @Size(min = 3, max = 50, message = "Name must be between 3 and 50 characters")  
 private String name;  
  
 @NotEmpty(message = "Email is required")  
 @Email(message = "Invalid email address")  
 private String email;  
  
 @NotEmpty(message = "Password is required")  
 @Size(min = 8, message = "Password must be at least 8 characters long")  
 private String password;  
  
 @NotNull(message = "Date of Birth is required")  
 @Past(message = "Date of Birth must be a past date")  
 private LocalDate dateOfBirth;  
  
 @NotEmpty(message = "Subscription plan is required")  
 private String subscriptionPlan;  
}

Model 2: Content

public class Content {  
  
 @NotEmpty(message = "ID cannot be empty")  
 @Size(min = 3, max = 20, message = "ID must be between 3 and 20 characters")  
 private String id;  
  
 @NotEmpty(message = "Title is required")  
 @Size(min = 3, max = 100, message = "Title must be between 3 and 100 characters")  
 private String title;  
  
 @NotEmpty(message = "Genre is required")  
 private String genre;  
  
 @NotNull(message = "Release date is required")  
 @PastOrPresent(message = "Release date must be a past or present date")  
 private LocalDate releaseDate;  
  
 @NotEmpty(message = "Content rating is required")  
 private String contentRating;  
  
 @Positive(message = "Duration must be a positive number")  
 private int duration; // in minutes  
  
 @NotEmpty(message = "Language is required")  
 private String language;  
}

Model 3: Subscription

public class Subscription {  
  
 @NotEmpty(message = "ID cannot be empty")  
 @Size(min = 3, max = 20, message = "ID must be between 3 and 20 characters")  
 private String id;  
  
 @NotEmpty(message = "User ID is required")  
 private String userId;  
  
 @NotEmpty(message = "Plan type is required")  
 private String planType;  
  
 @NotNull(message = "Start date is required")  
 @PastOrPresent(message = "Start date must be a past or present date")  
 private LocalDate startDate;  
  
 @NotNull(message = "End date is required")  
 @Future(message = "End date must be a future date")  
 private LocalDate endDate;  
  
 @NotNull(message = "Status is required")  
 private boolean status;  
}

Model 4: Transaction

public class Transaction {  
  
 @NotEmpty(message = "ID cannot be empty")  
 @Size(min = 3, max = 20, message = "ID must be between 3 and 20 characters")  
 private String id;  
  
 @NotEmpty(message = "User ID is required")  
 private String userId;  
  
 @NotNull(message = "Amount is required")  
 @Positive(message = "Amount must be a positive number")  
 private double amount;  
  
 @NotEmpty(message = "Payment method is required")  
 private String paymentMethod;  
  
 @NotNull(message = "Transaction date is required")  
 @PastOrPresent(message = "Transaction date must be a past or present date")  
 private LocalDate transactionDate;  
}