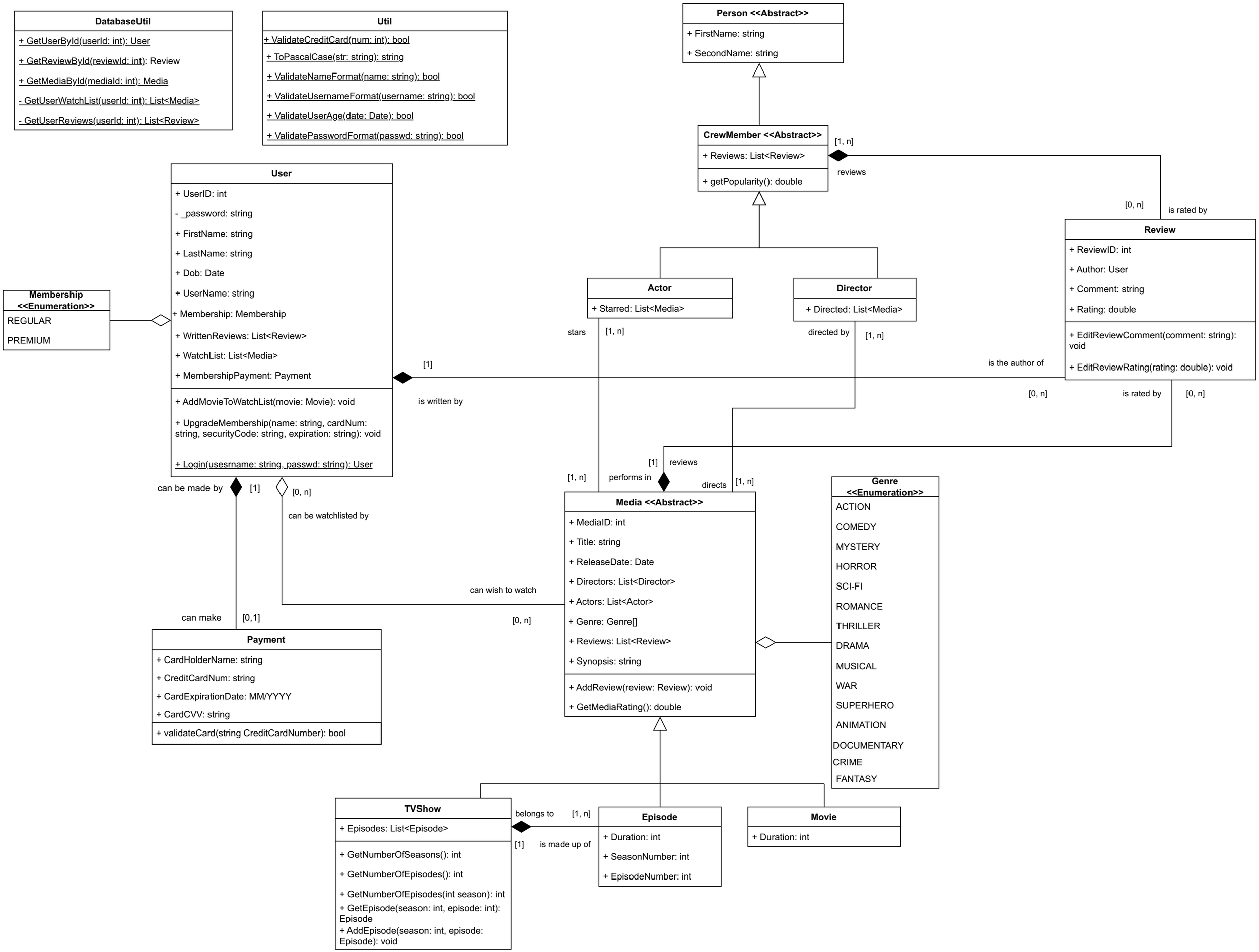
DatabaseUtil
+ GetUserById(userId: int): User
+ GetReviewById(reviewId: int): Review
+ GetMediaById(mediaId: int): Media
- GetUserWatchList(userId: int): List<Media>
- GetUserReviews(userId: int): List<Review>
Util
+ ValidateCreditCard(num: int): bool
+ ToPascalCase(str: string): string
+ ValidateNameFormat(name: string): bool
+ ValidateUsernameFormat(username: string): bool
+ ValidateUserAge(date: Date): bool
+ ValidatePasswordFormat(passwd: string): bool
Person <<Abstract>>
+ FirstName: string
+ SecondName: string
CrewMember <<Abstract>>
+ Reviews: List<Review>
+ getPopularity(): double
[1, n]
reviews
[0, n]
is rated by
Review
+ ReviewID: int
+ Author: User
+ Comment: string
+ Rating: double
+ EditReviewComment(comment: string): void
+ EditReviewRating(rating: double): void
Membership <<Enumeration>>
REGULAR
PREMIUM
User
+ UserID: int
- _password: string
+ FirstName: string
+ LastName: string
+ Dob: Date
+ UserName: string
+ Membership: Membership
+ WrittenReviews: List<Review>
+ WatchList: List<Media>
+ MembershipPayment: Payment
+ AddMovieToWatchList(movie: Movie): void
+ UpgradeMembership(name: string, cardNum: string, securityCode: string, expiration: string): void
+ Login(usesrname: string, passwd: string): User
Actor
+ Starred: List<Media>
Director
+ Directed: List<Media>
stars
[1, n]
directed by
[1, n]
[1]
is the author of
is written by
[0, n]
is rated by
[0, n]
[1]
reviews
[1, n]
performs in
directs
[1, n]
can be made by
[1]
[0, n]
can be watchlisted by
Media <<Abstract>>
+ MediaID: int
+ Title: string
+ ReleaseDate: Date
+ Directors: List<Director>
+ Actors: List<Actor>
+ Genre: Genre[]
+ Reviews: List<Review>
+ Synopsis: string
+ AddReview(review: Review): void
+ GetMediaRating(): double
Genre <<Enumeration>>
ACTION
COMEDY
MYSTERY
HORROR
SCI-FI
ROMANCE
THRILLER
DRAMA
MUSICAL
WAR
SUPERHERO
ANIMATION
DOCUMENTARY
CRIME
FANTASY
can wish to watch
[0, n]
can make
[0,1]
Payment
+ CardHolderName: string
+ CreditCardNum: string
+ CardExpirationDate: MM/YYYY
+ CardCVV: string
+ validateCard(string CreditCardNumber): bool
TVShow
+ Episodes: List<Episode>
+ GetNumberOfSeasons(): int
+ GetNumberOfEpisodes(): int
+ GetNumberOfEpisodes(int season): int
+ GetEpisode(season: int, episode: int): Episode
+ AddEpisode(season: int, episode: Episode): void
belongs to
[1, n]
[1]
is made up of
Episode
+ Duration: int
+ SeasonNumber: int
+ EpisodeNumber: int
Movie
+ Duration: int