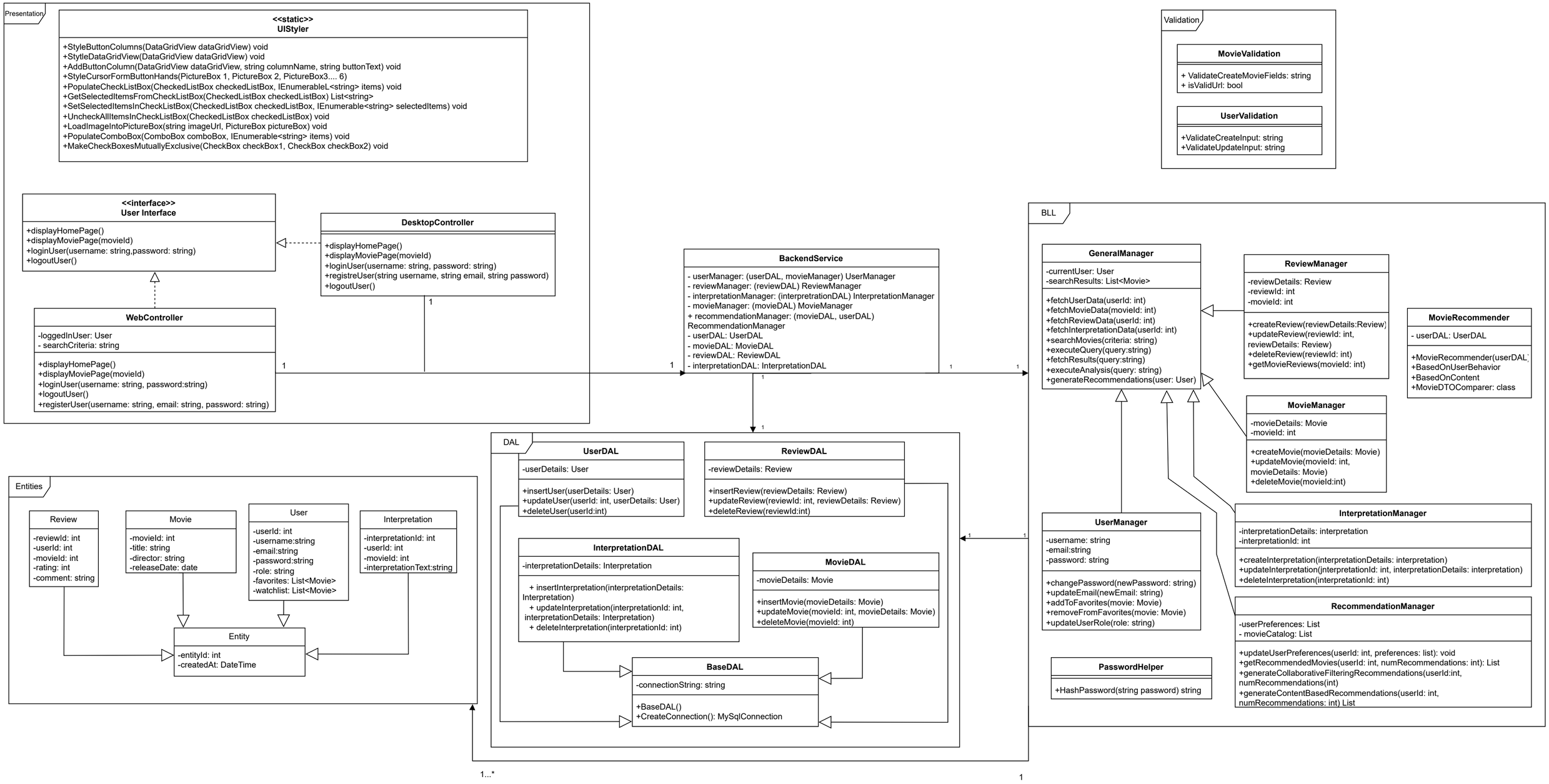
Presentation
<<static>>
UIStyler
+StyleButtonColumns(DataGridView dataGridView) void
+StytleDataGridView(DataGridView dataGridView) void
+AddButtonColumn(DataGridView dataGridView, string columnName, string buttonText) void
+StyleCursorFormButtonHands(PictureBox 1, PictureBox 2, PictureBox3.... 6)
+PopulateCheckListBox(CheckedListBox checkedListBox, IEnumerableL<string> items) void
+GetSelectedItemsFromCheckListBox(CheckedListBox checkedListBox) List<string>
+SetSelectedItemsInCheckListBox(CheckedListBox checkedListBox, IEnumerable<string> selectedItems) void
+UncheckAllItemsInCheckListBox(CheckedListBox checkedListBox) void
+LoadImageIntoPictureBox(string imageUrl, PictureBox pictureBox) void
+PopulateComboBox(ComboBox comboBox, IEnumerable<string> items) void
+MakeCheckBoxesMutuallyExclusive(CheckBox checkBox1, CheckBox checkBox2) void
<<interface>>
User Interface
+displayHomePage()
+displayMoviePage(movieId)
+loginUser(username: string,password: string)
+logoutUser()
DesktopController
+displayHomePage()
+displayMoviePage(movieId)
+loginUser(username: string, password: string)
+registreUser(string username, string email, string password)
+logoutUser()
WebController
-loggedInUser: User
- searchCriteria: string
+displayHomePage()
+displayMoviePage(movieId)
+loginUser(username: string, password:string)
+logoutUser()
+registerUser(username: string, email: string, password: string)
1
1
1
BackendService
- userManager: (userDAL, movieManager) UserManager
- reviewManager: (reviewDAL) ReviewManager
- interpretationManager: (interpretrationDAL) InterpretationManager
- movieManager: (movieDAL) MovieManager
+ recommendationManager: (movieDAL, userDAL) RecommendationManager
- userDAL: UserDAL
- movieDAL: MovieDAL
- reviewDAL: ReviewDAL
- interpretationDAL: InterpretationDAL
1
1
1
1
Validation
MovieValidation
+ ValidateCreateMovieFields: string
+ isValidUrl: bool
UserValidation
+ValidateCreateInput: string
+ValidateUpdateInput: string
BLL
GeneralManager
-currentUser: User
-searchResults: List<Movie>
+fetchUserData(userId: int)
+fetchMovieData(movieId: int)
+fetchReviewData(userId: int)
+fetchInterpretationData(userId: int)
+searchMovies(criteria: string)
+executeQuery(query:string)
+fetchResults(query:string)
+executeAnalysis(query: string)
+generateRecommendations(user: User)
ReviewManager
-reviewDetails: Review
-reviewId: int
-movieId: int
+createReview(reviewDetails:Review)
+updateReview(reviewId: int, reviewDetails: Review)
+deleteReview(reviewId: int)
+getMovieReviews(movieId: int)
MovieRecommender
- userDAL: UserDAL
+MovieRecommender(userDAL)
+BasedOnUserBehavior
+BasedOnContent
+MovieDTOComparer: class
MovieManager
-movieDetails: Movie
-movieId: int
+createMovie(movieDetails: Movie)
+updateMovie(movieId: int, movieDetails: Movie)
+deleteMovie(movieId:int)
UserManager
-username: string
-email:string
-password: string
+changePassword(newPassword: string)
+updateEmail(newEmail: string)
+addToFavorites(movie: Movie)
+removeFromFavorites(movie: Movie)
+updateUserRole(role: string)
InterpretationManager
-interpretationDetails: interpretation
-interpretationId: int
+createInterpretation(interpretationDetails: interpretation)
+updateInterpretation(jinterpretationId: int, interpretationDetails: interpretation)
+deleteInterpretation(interpretationId: int)
RecommendationManager
-userPreferences: List
- movieCatalog: List
+updateUserPreferences(userId: int, preferences: list): void
+getRecommendedMovies(userId: int, numRecommendations: int): List
+generateCollaborativeFilteringRecommendations(userId:int, numRecommendations(int)
+generateContentBasedRecommendations(userId: int, numRecommendations: int) List
PasswordHelper
+HashPassword(string password) string
1
1
DAL
UserDAL
-userDetails: User
+insertUser(userDetails: User)
+updateUser(userId: int, userDetails: User)
+deleteUser(userId:int)
ReviewDAL
-reviewDetails: Review
+insertReview(reviewDetails: Review)
+updateReview(reviewId: int, reviewDetails: Review)
+deleteReview(reviewId:int)
InterpretationDAL
-interpretationDetails: Interpretation
+ insertInterpretation(interpretationDetails: Interpretation)
+ updateInterpretation(interpretationId: int, interpretationDetails: Interpretation)
+ deleteInterpretation(interpretationId: int)
MovieDAL
-movieDetails: Movie
+insertMovie(movieDetails: Movie)
+updateMovie(movieId: int, movieDetails: Movie)
+deleteMovie(movieId: int)
BaseDAL
-connectionString: string
+BaseDAL()
+CreateConnection(): MySqlConnection
Entities
Review
-reviewId: int
-userId: int
-movieId: int
-rating: int
-comment: string
Movie
-movieId: int
-title: string
-director: string
-releaseDate: date
User
-userId: int
-username:string
-email:string
-password:string
-role: string
-favorites: List<Movie>
-watchlist: List<Movie>
Interpretation
-interpretationId: int
-userId: int
-movieId: int
-interpretationText:string
Entity
-entityId: int
-createdAt: DateTime
1...*
1