# Model Class Diagram without Relationships, and with Attributes and Behaviour
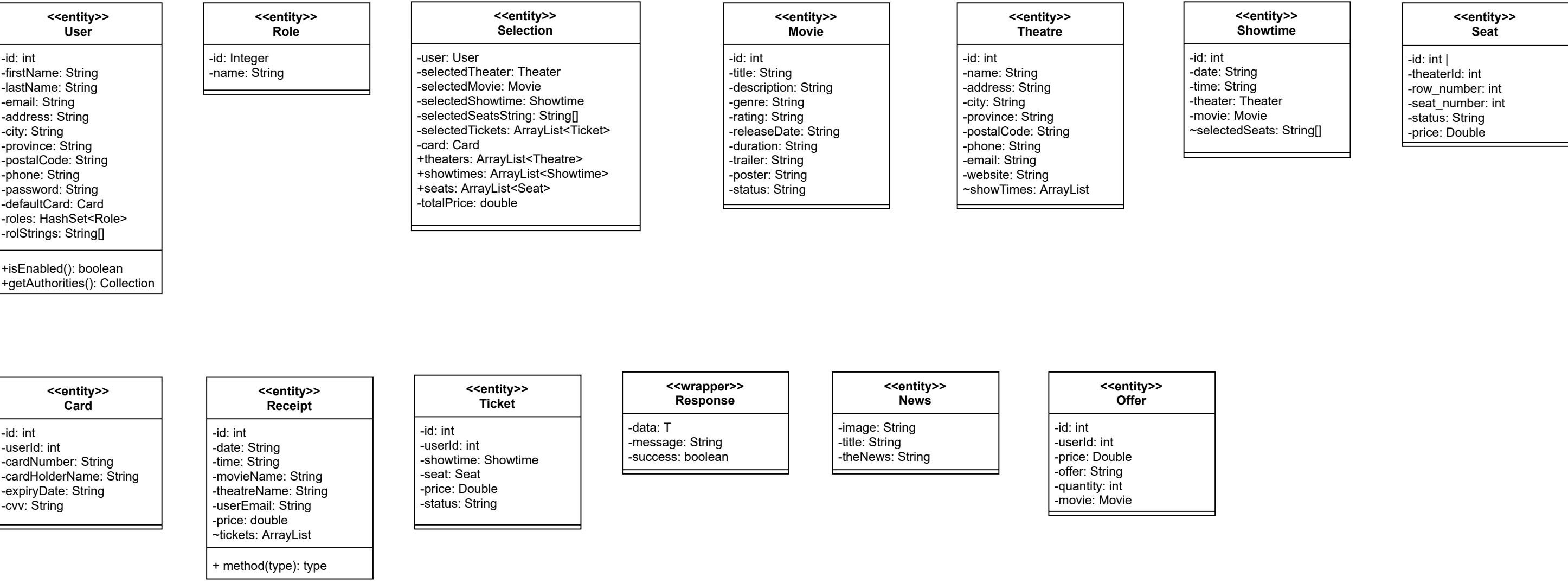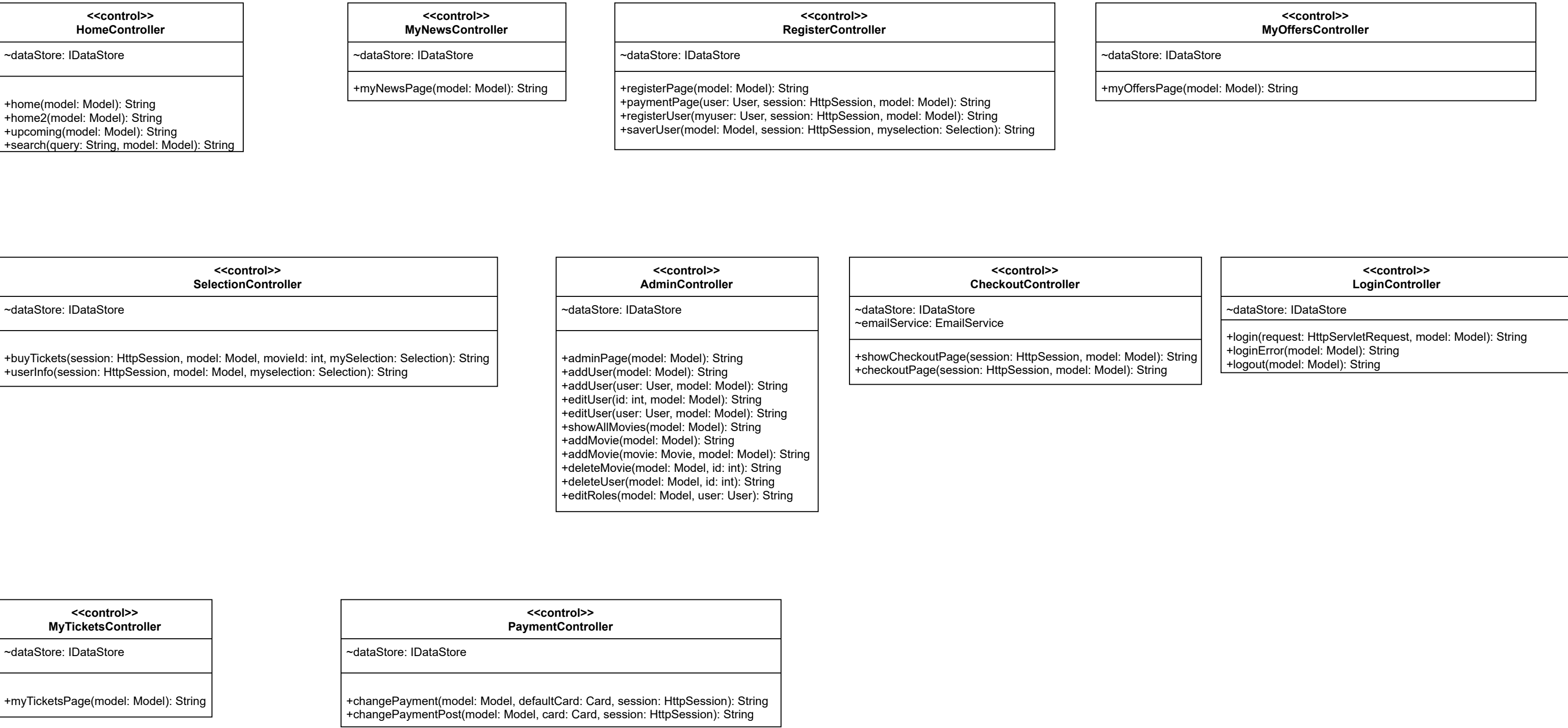
Movie Ticket Reservation App

Group 5

---

**<<entity>>**
**User**

- -id: int
- -firstName: String
- -lastName: String
- -email: String
- -address: String
- -city: String
- -province: String
- -postalCode: String
- -phone: String
- -password: String
- -defaultCard: Card
- -roles: HashSet<Role>
- -rolStrings: String[]

+isEnabled(): boolean
+getAuthorities(): Collection

---

**<<entity>>**
**Role**

- -id: Integer
- -name: String

---

**<<entity>>**
**Selection**

- -user: User
- -selectedTheater: Theater
- -selectedMovie: Movie
- -selectedShowtime: Showtime
- -selectedSeatsString: String[]
- -selectedTickets: ArrayList<Ticket>
- -card: Card
- +theaters: ArrayList<Theatre>
- +showtimes: ArrayList<Showtime>
- +seats: ArrayList<Seat>
- +totalPrice: double

---

**<<entity>>**
**Movie**

- -id: int
- -title: String
- -description: String
- -genre: String
- -rating: String
- -releaseDate: String
- -duration: String
- -trailer: String
- -poster: String
- -status: String

---

**<<entity>>**
**Theatre**

- -id: int
- -name: String
- -address: String
- -city: String
- -province: String
- -postalCode: String
- -phone: String
- -email: String
- -website: String
- ~showTimes: ArrayList

---

**<<entity>>**
**Showtime**

- -id: int
- -date: String
- -time: String
- -theater: Theater
- -movie: Movie
- ~selectedSeats: String[]

---

**<<entity>>**
**Seat**

- -id: int
- -theaterId: int
- -row_number: int
- -seat_number: int
- -status: String
- -price: Double

---

**<<entity>>**
**Card**

- -id: int
- -userId: int
- -cardNumber: String
- -cardHolderName: String
- -expiryDate: String
- -cvv: String

---

**<<entity>>**
**Receipt**

- -id: int
- -date: String
- -time: String
- -movieName: String
- -theatreName: String
- -userEmail: String
- -price: double
- ~tickets: ArrayList

+ method(type): type

---

**<<entity>>**
**Ticket**

- -id: int
- -userId: int
- -showtime: Showtime
- -seat: Seat
- -price: Double
- -status: String

---

**<<wrapper>>**
**Response**

- -data: T
- -message: String
- -success: boolean

---

**<<entity>>**
**News**

- -image: String
- -title: String
- -theNews: String

---

**<<entity>>**
**Offer**

- -id: int
- -userId: int
- -price: Double
- -offer: String
- -quantity: int
- -movie: Movie

---

# Controller Class Diagram without Relationships, and with Attributes and Behaviour

**<<control>>**
**HomeController**

~dataStore: IDataStore

+home(model: Model): String
+home2(model: Model): String
+upcoming(model: Model): String
+search(query: String, model: Model): String

---

**<<control>>**
**MyNewsController**

~dataStore: IDataStore

+myNewsPage(model: Model): String

---

**<<control>>**
**RegisterController**

~dataStore: IDataStore

+registerPage(model: Model): String
+paymentPage(user: User, session: HttpSession, model: Model): String
+registerUser(myuser: User, session: HttpSession, model: Model): String
+saverUser(model: Model, session: HttpSession, myselection: Selection): String

---

**<<control>>**
**MyOffersController**

~dataStore: IDataStore

+myOffersPage(model: Model): String

---

**<<control>>**
**SelectionController**

~dataStore: IDataStore

+buyTickets(session: HttpSession, model: Model, movieId: int, mySelection: Selection): String
+userInfo(session: HttpSession, model: Model, myselection: Selection): String

---

**<<control>>**
**AdminController**

~dataStore: IDataStore

+adminPage(model: Model): String
+addUser(model: Model): String
+addUser(user: User, model: Model): String
+editUser(id: int, model: Model): String
+editUser(user: User, model: Model): String
+showAllMovies(model: Model): String
+addMovie(model: Model): String
+addMovie(movie: Movie, model: Model): String
+deleteMovie(model: Model, id: int): String
+deleteUser(model: Model, id: int): String
+editRoles(model: Model, user: User): String

---

**<<control>>**
**CheckoutController**

~dataStore: IDataStore
~emailService: EmailService

+showCheckoutPage(session: HttpSession, model: Model): String
+checkoutPage(session: HttpSession, model: Model): String

---

**<<control>>**
**LoginController**

~dataStore: IDataStore

+login(request: HttpServletRequest, model: Model): String
+loginError(model: Model): String
+logout(model: Model): String

---

**<<control>>**
**MyTicketsController**

~dataStore: IDataStore

+myTicketsPage(model: Model): String

---

**<<control>>**
**PaymentController**

~dataStore: IDataStore

+changePayment(model: Model, defaultCard: Card, session: HttpSession): String
+changePaymentPost(model: Model, card: Card, session: HttpSession): String

---

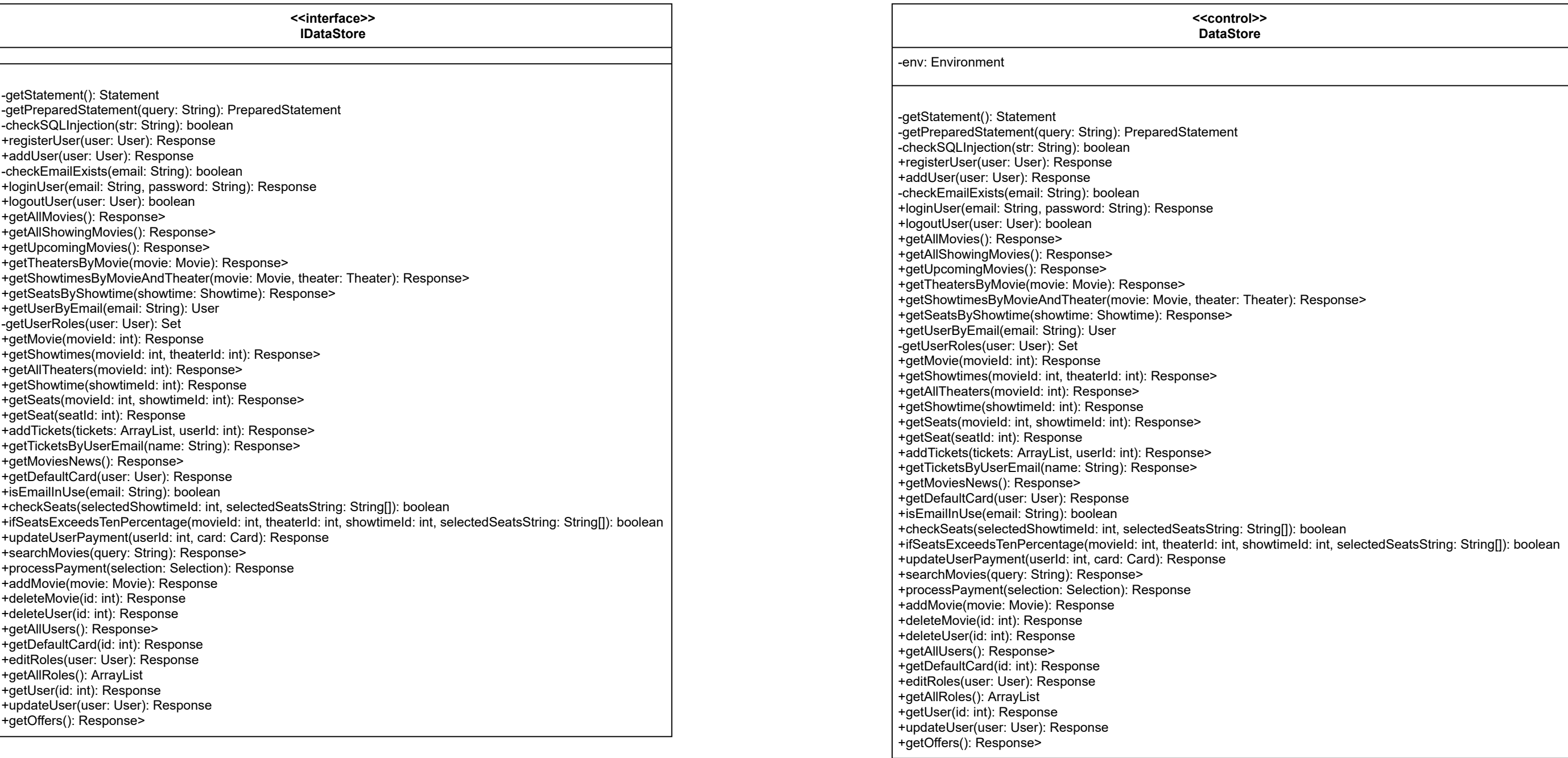# Data Control Class Diagram without Relationships, and with Attributes and Behaviour

**<<interface>>**
**IDataStore**

-getStatement(): Statement
-getPreparedStatement(query: String): PreparedStatement
-checkSQLInjection(str: String): boolean
+registerUser(user: User): Response
+addUser(user: User): Response
-checkEmailExists(email: String): boolean
+loginUser(email: String, password: String): Response
+logoutUser(user: User): boolean
+getAllMovies(): Response>
+getAllShowingMovies(): Response>
+getUpcomingMovies(): Response>
+getTheatersByMovie(movie: Movie): Response>
+getShowtimesByMovieAndTheater(movie: Movie, theater: Theater): Response>
+getSeatsByShowtime(showtime: Showtime): Response>
+getUserByEmail(email: String): User
+getUserRoles(user: User): Set
+getMovie(movieId: int): Response
+getShowtimes(movieId: int, theaterId: int): Response>
+getAllTheaters(movieId: int): Response>
+getShowtime(showtimeId: int): Response
+getSeats(movieId: int, showtimeId: int): Response>
+getSeat(seatId: int): Response
+addTickets(tickets: ArrayList, userId: int): Response>
+getTicketsByUserEmail(name: String): Response>
+getMoviesNews(): Response>
+getDefaultCard(user: User): Response
+isEmailInUse(email: String): boolean
+checkSeats(selectedShowtimeId: int, selectedSeatsString: String[]): boolean
+ifSeatsExceedsTenPercentage(movieId: int, theaterId: int, showtimeId: int, selectedSeatsString: String[]): boolean
+updateUserPayment(userId: int, card: Card): Response
+searchMovies(query: String): Response
+processPayment(selection: Selection): Response
+addMovie(movie: Movie): Response
+deleteMovie(id: int): Response
+deleteUser(id: int): Response
+getAllUsers(): Response>
+getDefaultCard(id: int): Response
+editRoles(user: User): Response
+getAllRoles(): ArrayList
+getUser(id: int): Response
+updateUser(user: User): Response
+getOffers(): Response>

---

**<<control>>**
**DataStore**

-env: Environment

-getStatement(): Statement
-getPreparedStatement(query: String): PreparedStatement
-checkSQLInjection(str: String): boolean
+registerUser(user: User): Response
+addUser(user: User): Response
-checkEmailExists(email: String): boolean
+loginUser(email: String, password: String): Response
+logoutUser(user: User): boolean
+getAllMovies(): Response>
+getAllShowingMovies(): Response>
+getUpcomingMovies(): Response>
+getTheatersByMovie(movie: Movie): Response>
+getShowtimesByMovieAndTheater(movie: Movie, theater: Theater): Response>
+getSeatsByShowtime(showtime: Showtime): Response>
+getUserByEmail(email: String): User
+getUserRoles(user: User): Set
+getMovie(movieId: int): Response
+getShowtimes(movieId: int, theaterId: int): Response>
+getAllTheaters(movieId: int): Response>
+getShowtime(showtimeId: int): Response
+getSeats(movieId: int, showtimeId: int): Response>
+getSeat(seatId: int): Response
+addTickets(tickets: ArrayList, userId: int): Response>
+getTicketsByUserEmail(name: String): Response>
+getMoviesNews(): Response>
+getDefaultCard(user: User): Response
+isEmailInUse(email: String): boolean
+checkSeats(selectedShowtimeId: int, selectedSeatsString: String[]): boolean
+ifSeatsExceedsTenPercentage(movieId: int, theaterId: int, showtimeId: int, selectedSeatsString: String[]): boolean
+updateUserPayment(userId: int, card: Card): Response
+searchMovies(query: String): Response>
+processPayment(selection: Selection): Response
+addMovie(movie: Movie): Response
+deleteMovie(id: int): Response
+deleteUser(id: int): Response
+getAllUsers(): Response>
+getDefaultCard(id: int): Response
+editRoles(user: User): Response
+getAllRoles(): ArrayList
+getUser(id: int): Response
+updateUser(user: User): Response
+getOffers(): Response>

---

# View Class Diagram without Relationships, and with Attributes and Behaviour

**<<boundary>>** about.html

**<<boundary>>** contact.html

**<<boundary>>** myticketsguest.html

**<<boundary>>** navbar.html

**<<boundary>>** userinfo.html

**<<boundary>>** upcoming.html

**<<boundary>>** select.html

**<<boundary>>** registration_payment.html

**<<boundary>>** register.html

**<<boundary>>** edituser.html

**<<boundary>>** mytickets.html

**<<boundary>>** myoffers.html

**<<boundary>>** mynews.html

**<<boundary>>** login.html

**<<boundary>>** home.html

**<<boundary>>** head.html

**<<boundary>>** footer.html

**<<boundary>>** error.html

**<<boundary>>** addmovie.html

**<<boundary>>** confirmation.html

**<<boundary>>** checkout.html

**<<boundary>>** changepayment.html

**<<boundary>>** cancelconfirm.html

**<<boundary>>** cancel.html

**<<boundary>>** adminmovies.html

**<<boundary>>** admin.html

**<<boundary>>** adduser.html