

InputOutputHandler
- cinemaService: CinemaService - instance: InputOutputHandler
+ editBooking(Booking, User): Page - readInput(int): int + seatsSelectionPage(ShowTime, Booking): List<Seat> + movieSelectionPage(Cinema): Movie + confirmPaymentPage(Booking, User, Booking): boolean - chooseOption(List<String>, String): int - changeUserData(User, int): Page + rechargeAccount(User): boolean + homePage(boolean): Page + loginOrRegisterPage(): User + bookingManagePage(User): Booking - refundUser(Booking, User): boolean + accountManagementPage(User): Page + getInstance(): InputOutputHandler + cinemaSelectionPage(): Cinema + editAccount(User): Page + getInstance(CinemaService): InputOutputHandler + showTimeSelectionPage(Movie, Cinema): ShowTime

CinemaDatabase
+ DB_URL: String - inTransaction: boolean # connection: Connection
- connect(String): Connection + withTransaction(ThrowingRunnable): void + getConnection(String): Connection + isInTransaction(): boolean + isDatabaseEmpty(): boolean

Main
+ main(String[]): void - populateDatabase(): void - getSomeSeats(): List<Seat>

HallFactory
+ createHall(Hall): Hall + createHall(ResultSet): Hall

<<interface>> CinemaService
+ pay(Booking, Booking?, User): void + addShowTime(ShowTime): void + register(String, String): User + retrieveCinemas(): List<Cinema> + retrieveBookings(User): List<Booking> + addMovie(Movie): void + updateUser(User, String, String): void + addSeat(Seat, Hall): void + retrieveCinemaMovies(Cinema): List<Movie> + login(String, String): User + addUser(User): void + addMovie(Movie, Hall, LocalDateTime): void + deleteBooking(Booking, User): void + addHall(Hall): void + retrieveShowTimeHallSeats(ShowTime): List<Seat> + retrieveMovieShowTimes(Movie, Cinema): List<ShowTime> + deleteUser(User): void + addCinema(Cinema): void + rechargeAccount(User, long): void

CinemaServiceImpl
- cinemaRepo: CinemaRepository - seatRepo: SeatRepository - userRepo: UserRepository - showTimeRepo: ShowTimeRepository - bookingRepo: BookingRepository - hallRepo: HallRepository - instance: CinemaService - movieRepo: MovieRepository
+ updateUser(User, String, String): void + retrieveMovieShowTimes(Movie, Cinema): List<ShowTime> + login(String, String): User + getInstance(): CinemaService + addCinema(Cinema): void + retrieveCinemas(): List<Cinema> + retrieveBookings(User): List<Booking> + rechargeAccount(User, long): void + register(String, String): User + addMovie(Movie, Hall, LocalDateTime): void + pay(Booking, Booking?, User): void + addSeat(Seat, Hall): void + deleteBooking(Booking, User): void + addHall(Hall): void + addUser(User): void + retrieveCinemaMovies(Cinema): List<Movie> + getInstance(CinemaRepository, HallRepository, MovieRepository, ShowTimeRepository): CinemaServiceImpl - encryptPassword(String): String + addShowTime(ShowTime): void + retrieveShowTimeHallSeats(ShowTime): List<Seat> + deleteUser(User): void + addMovie(Movie): void