

GlobalModelAttributes
+ addGlobalAttributes(HttpSession, Model): void

HomeController
+ showHomePage(): String

UserController
- service: UserService
+ showLoginForm(HttpSession): String
+ login(String, String, HttpSession, Model): String
+ logout(HttpSession): String
+ showRegistrationForm(HttpSession, Model): String
+ registerUser(User, BindingResult, String, Model): String
+ showUserAdminPage(HttpSession, Model): String
+ updateUser (User, BindingResult, HttpSession, Model): String
+ showChangePasswordForm(HttpSession): String
+ changePassword(String, String, String, HttpSession, Model) : String
+ deleteUser(HttpSession, Model): String

UserService
- repository: UserRepository
+ authenticate(String, String): User
+ usernameExists(String): boolean
+ emailExists(String): boolean
+ registerUser(User): boolean
+ getUserByUsername(String): User
+ updateUser(String, User): boolean
+ changePassword(String, String): boolean
+ deleteUser(String): boolean

UserRepository
- jdbcTemplate: JdbcTemplate
+ getUserByUsername(String): User
+ getUserByEmail(String): User
+ getUserRowMapper(): RowMapper<User>
- usernameExists(String): String
- emailExists(String): String
+ registerUser(User): boolean
+ updateUser(String, User): boolean
+ changePassword(String, String): boolean
+ deleteUser(String) : boolean

User
- username: String
- password: String
- email: String
+ getter and setters

WishList
- id: int
- username: String
- title: String
- description: String
- eventDate: LocalDate
- notPublic: boolean
+ getter and setters

WishItem
- id: int
- wishListId: int
- title: String
- favourite: boolean
- description: String
- price: double
- quantity: int
- link: String
- reserved: boolean
- reservedBy: String
+ getter and setters

WishListController
- service: WishListService
+ getUserWishLists(HttpSession, Model): String
+ getWishList(String, HttpSession, Model): String
+ getWishItem(String, HttpSession Model): String
+ showCreateWishListForm(HttpSession, Model): String
+ createWishList(HttpSession, WishList, BindingResult, Model): String
+ showAddWishForm(String, HttpSession, Model): String
+ addWishItem(String, HttpSession, WishItem, BindingResult, Model): String
+ showEditWishListForm (String, HttpSession, Model): String
+ updateWishList (HttpSession, WishList, BindingResult, Model): String
+ showEditWishItemForm(String, HttpSession, Model): String
+ updateWishItem(HttpSession, WishItem, BindingResult, Model): String
+ deleteWishList (HttpSession, String, RedirectAttributes): String
+ deleteWishItem(HttpSession, String, Model): String
+ reserveWishItem(HttpSession, String, RedirectAttributes): String
+ unreserveWishItem(HttpSession, String, RedirectAttributes): String
- getCurrentUsername(HttpSession): String
- isOwner(HttpSession, WishList): boolean
- getWishListIfOwner(String, HttpSession): WishList
- addOwnershipToModel(HttpSession, Model, WishList): void

WishListService
- repository: WishListRepository
+ getUserWishLists(String): List<WishList>
+ getWishList(String): WishList
+ getWishListItems(String): List<WishItem>
+ getWishItem(String): WishItem
+ createWishListAndReturnId(WishList): int
+ addWishItem(WishItem): boolean
+ updateWishList(WishList): boolean
+ updateWishItem(WishItem): boolean
+ deleteWishList(String): boolean
+ deleteWishItem(String): boolean

WishListRepository
- jdbcTemplate: JdbcTemplate
+ getUserWishLists(String): List<WishList>
+ getWishList(String): WishList
+ getWishListItems(String): List<WishItem>
+ getWishItem(String): WishItem
+ createWishListAndReturnId(WishList): int
+ addWishItem(WishItem): boolean
+ updateWishList(WishList): boolean
+ updateWishItem(WishItem): boolean
+ deleteWishList(String): boolean
+ deleteWishItem(String): boolean
- getWishListRowMapper(): RowMapper<WishList>
- getWishItemRowMapper: RowMapper<WishItem>

SessionUtil
+ isLoggedIn(HttpSession): boolean

FormatUtil
+ isValidEmail(String): boolean