# **Attributes UML**

# ManagementSystemController

- PORT: int
- serverSocket: ServerSocket
- databaseController: Database(
- pool: ExecutorService
- + communicateWithClient(): voic
- + printlPInfo(): void

# <<interface>> DatabaseCredentials

- + JDBC\_DRIVER: String
- + DB\_URL: String + USERNAME: String
- + PASSWORD: String

# <<!nterface>> Observer

+ update(newListing: Listing): void

# Manager

- + getPeriodicalReport(): PeriodicalRepo
- + getEntityInfo(): Entity

## RegisteredRenter

- databaseModel: DatabaseModel
- + update(listing: Listing): void

# **ServerCommunicationController**

- aSocket: Socket
- socketIn: ObjectInputStream
- socketOut: ObjectOutputStream
- serverController: ServerController
- + run(): void @override
- + communicate(): void
- + sendListingsToClient(): void
- + sendEntityToClient(): void
- + verifyLogin(): void
- + createUniqueInputStream(): void
- + addListingToDB(): void
- + removeListingFromDB(): void
- + editListingInDB(): void

# <<Interface>>

### **DatabaseAccessQueries**

- + SQL\_GET\_LISTINGS: String
- + SQL\_GET\_LISTING: String
- + SQL\_GET\_USER: String
- + SQL\_EDIT\_LISTING: String
- + SQL\_ADD\_LISTING: String
- + SQL\_ADD\_USER: String
- + SQL\_REMOVE\_USER: String
- + SQL\_REMOVE\_LISTING: String

### **DatabaseController**

- myConnection: Connection
- databaseModel: DatabaseModel
- + initializeConnection(): void
- + closeConnection(): void

# <<Interface>> Observable

- + addObserver(Observer o): void
- + removeObserver(Observer o): void
- + notifyAllObservers(): void

### User

- # username: String # password: String
- # email: String # identity: String
- + compareUser(User: user): boolean
- + getIdentity(): String

# DatabaseModel

- myConnection: Connection
- observers: ArrayList<Observer>
- + verifyUser(user: User): boolean
- + addUser(user: User): void
- + getListingsFromDB(): ArrayList<Listing>
- + getUserFromDB(): Üser
- + editListingInDB(Listing listing): void
- + removeListingFromDB(Listing listing): void
- + addObserver(Observer o): void
- + removeObserver(Observer o): void
- + notifyAllObservers(): void
- + addListingToDB(): void

### Landlord

- listings: ArrayList<Listing>
- + addListing(listing: Listing): void
- + removeListing(listing: Listing): void

#### ClientCommunicationController

- socketOut: ObjectOutputStream
- aSocket: Socket
- socketIn: ObjectInputStreamloginController: ControlleremailController: Controller
- mainController: ControllersearchListingController: Controller
- listingsController: Controller
- createListingController: Controller
- + main(args: String[]): void
- + showMainWindow(): void + showEmailView(): void
- + showSearchListingView(): void
- + showListingsView(): void
- + showCreateListingView(): void

# LoginController

- loginView: LoginViewverified: booleanuser: User
- + loginListen(): void + isVerified(): boolean

#### Controller

# clientCommunicationController: ClientCommunicationController

#### **EmailController**

- emailView: EmailView
- + sendEmailListen();

# SearchListingController

- searchListingView: SearchListingView
- + searchListingListen();

# ListingsController

- listingsView: ListingsView
- + selectListingListen();

#### MainController

- mainView: MainView
- + searchListingListen(): void
- + editListingListen(): void
- + unsubscribeListen(): void
- + createListingListen(): void
- + getPeriodicalReportListen(): void
- + changeFeeListen();
- + viewEntityListen();

### **PeriodicalReportView**

- components: Components
- + display(): void
- + hide(): void

# PeriodicalReportController

- periodicalReportView: PeriodicalReportView
- + createReportListen();

## PeriodicalReport

- numOfHousesListed: int
- numOfHousesRented: int
- numOfActiveListings: int
- listings: ArrayList<Listing>
- startDate: Date
- endDate: Date

# ListingsView

- components: Components
- + display(): void
- + hide(): void

## **EmailView**

- components: Components
- + display(): void
- + hide(): void

## SearchListingView

- components: Components
- + display(): void
- + hide(): void

# CreateListingView

- components: Components
- + display(): void
- + hide(): void