

Model::Main

```
classDiagram
    class UserInterfaceSignUpUI {
        <<boundary>>
        -control_ : SignUp&
        +startInterface() void
    }
    class UserInterfaceLoginUI {
        <<boundary>>
        -control_ : Login&
        +startInterface() void
    }
    class UserInterfaceLogoutUI {
        <<boundary>>
        -control_ : Logout&
        +startInterface() void
    }
    class UserInterfaceAddBicycleUI {
        <<boundary>>
        -control_ : AddBicycle&
        +startInterface() void
    }
    class UserInterfaceViewRentInfoUI {
        <<boundary>>
        -control_ : ViewRentInfo&
        +startInterface() void
    }
    class UserInterfaceRentBicycleUI {
        <<boundary>>
        -control_ : RentBicycle&
        +startInterface() void
    }
    class ControlSignUp {
        <<control>>
        +addNewUser(id: String, password: String, phoneNumber: String): Boolean
    }
    class ControlLogin {
        <<control>>
        +doLogin(userID: String, password: String): Boolean
    }
    class ControlLogout {
        <<control>>
        +doLogout(outUserID: String): Boolean
    }
    class ControlViewRentInfo {
        <<control>>
        +getRentInfo(): vector<pair<string,string>>
    }
    class ControlRentBicycle {
        <<control>>
        +rentBicycle(bicycleID: String): String
    }
    class ControlAddBicycle {
        <<control>>
        +addNewBicycle(bicycleID: String, bicycleName: String): Boolean
    }
    class UserList {
        <<entity>>
        -members: vector<Member>
        -currentUser: Member*
        +addNewMember(id: String, password: String, phoneNumber: String): void
        +validateUser(id: String, password: String): Boolean
        +getCurrentUser(): Member
        +getInstance(): UserList&
        +clearCurrentUser(): void
        +setCurrentUser(member: Member): void
    }
    class Member {
        <<entity>>
        -userID: String
        -password: String
        -phoneNumber: String
        +getID(): String
        +getPassword(): String
        +getPhoneNumber(): String
    }
    class BicycleList {
        <<entity>>
        -bicycles: vector<Bicycle>
        +addNewBicycle(id: String, name: String): void
        +findBicycle(id: String): Bicycle*
        +getAllBicycles(): vector<Bicycle>&
    }
    class Bicycle {
        <<entity>>
        -bicycleID: String
        -bicycleName: String
        -rentStatus: Boolean
        -renteduserID: String
        +getRentFor(): String
        +rentTo(userID: String): Boolean
        +getName(): String
        +getID(): String
    }
    UserList "1" -- "1..*" Member : has a
    BicycleList "1" -- "1..*" Bicycle : has a
    ControlSignUp --> UserInterfaceSignUpUI
    ControlLogin --> UserInterfaceLoginUI
    ControlLogout --> UserInterfaceLogoutUI
    ControlViewRentInfo --> UserInterfaceViewRentInfoUI
    ControlRentBicycle --> UserInterfaceRentBicycleUI
    ControlAddBicycle --> UserInterfaceAddBicycleUI
```

The diagram illustrates the structure of a bicycle rental system, organized into three main layers: Boundary, Control, and Entity.

Boundary Layer (User Interfaces):

- User Interface::SignUpUI**: Contains `-control_ : SignUp&` and `+startInterface(): void`.
- User Interface::LoginUI**: Contains `-control_ : Login&` and `+startInterface(): void`.
- User Interface::LogoutUI**: Contains `-control_ : Logout&` and `+startInterface(): void`.
- User Interface::AddBicycleUI**: Contains `-control_ : AddBicycle&` and `+startInterface(): void`.
- User Interface::ViewRentInfoUI**: Contains `-control_ : ViewRentInfo&` and `+startInterface(): void`.
- User Interface::RentBicycleUI**: Contains `-control_ : RentBicycle&` and `+startInterface(): void`.

Control Layer (Business Logic):

- Control::SignUp**: Contains `+addNewUser(id: String, password: String, phoneNumber: String): Boolean`.
- Control::Login**: Contains `+doLogin(userID: String, password: String): Boolean`.
- Control::Logout**: Contains `+doLogout(outUserID: String): Boolean`.
- Control::ViewRentInfo**: Contains `+getRentInfo(): vector<pair<string,string>>`.
- Control::RentBicycle**: Contains `+rentBicycle(bicycleID: String): String`.
- Control::AddBicycle**: Contains `+addNewBicycle(bicycleID: String, bicycleName: String): Boolean`.

Entity Layer (Data Objects):

- UserList**: Contains `-members: vector<Member>`, `-currentUser: Member*`, and methods `+addNewMember(id: String, password: String, phoneNumber: String): void`, `+validateUser(id: String, password: String): Boolean`, `+getCurrentUser(): Member`, `+getInstance(): UserList&`, `+clearCurrentUser(): void`, and `+setCurrentUser(member: Member): void`.
- Member**: Contains attributes `-userID: String`, `-password: String`, and `-phoneNumber: String`, and methods `+getID(): String`, `+getPassword(): String`, and `+getPhoneNumber(): String`.
- BicycleList**: Contains `-bicycles: vector<Bicycle>` and methods `+addNewBicycle(id: String, name: String): void`, `+findBicycle(id: String): Bicycle*`, and `+getAllBicycles(): vector<Bicycle>&`.
- Bicycle**: Contains attributes `-bicycleID: String`, `-bicycleName: String`, `-rentStatus: Boolean`, and `-renteduserID: String`, and methods `+getRentFor(): String`, `+rentTo(userID: String): Boolean`, `+getName(): String`, and `+getID(): String`.

Associations:

- UserList** (1) **has a** **Member** (1..*)
- BicycleList** (1) **has a** **Bicycle** (1..*)

Relationships:

- Control::SignUp** is associated with **User Interface::SignUpUI**.
- Control::Login** is associated with **User Interface::LoginUI**.
- Control::Logout** is associated with **User Interface::LogoutUI**.
- Control::ViewRentInfo** is associated with **User Interface::ViewRentInfoUI**.
- Control::RentBicycle** is associated with **User Interface::RentBicycleUI**.
- Control::AddBicycle** is associated with **User Interface::AddBicycleUI**.