Model::Collaboration2::Interaction1::Login

sd Login

: 유저

2 : enterCredentials()

: LoginUI

3 : ValidateCredential()

: Login

1 : startInterface()

4 : showLoginCredential()

3.1 : authenticateMember()

3.2 : setCurrentMember()

: Member

: MemberCollection
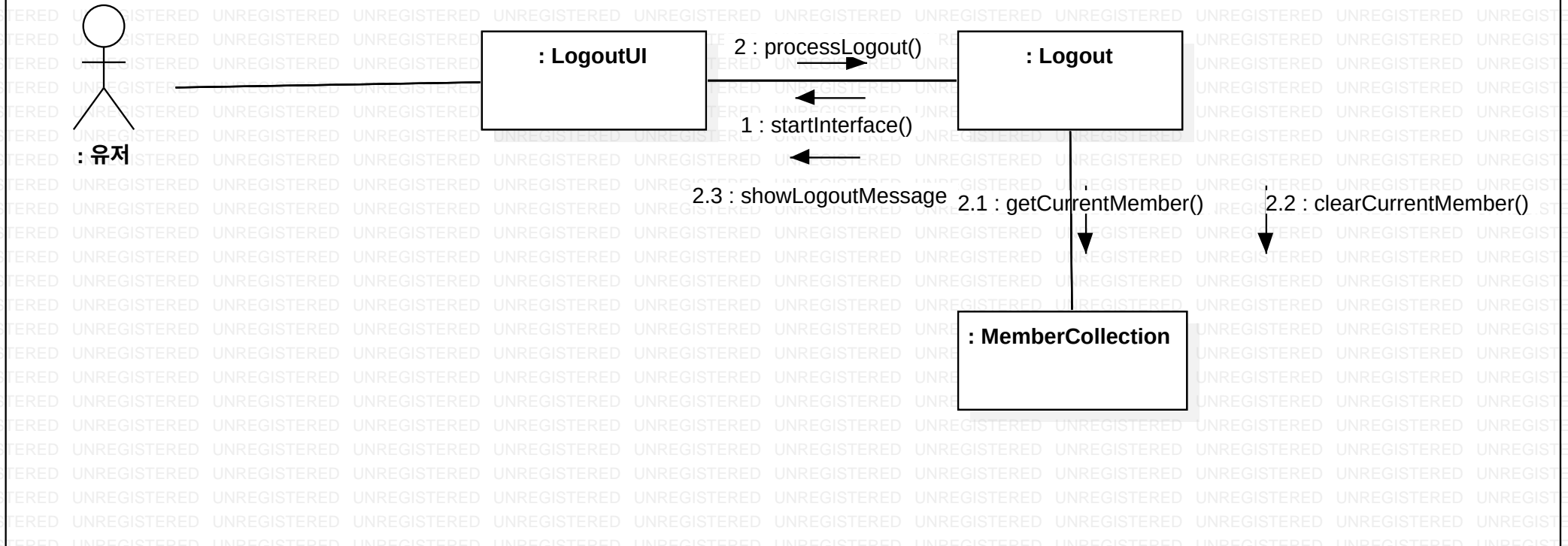
3.1.1 : checkMember()

sd Logout

: 유저

: LogoutUI

2 : processLogout()

1 : startInterface()

: Logout

2.3 : showLogoutMessage

2.1 : getCurrentMember()

2.2 : clearCurrentMember()

: MemberCollection

sd RegisterBike

: 관리자

2 : enterBicycleInfo()

: RegisterBicycleUI

2.1 : submitNewBicycle()

1 : startInterface()

2.2 : showRegisteredBicycle()

: RegisterBicycle

2.1.1 : Bicycle()

: BicycleCollection

: Bicycle

2.1.2 : addBicycle()
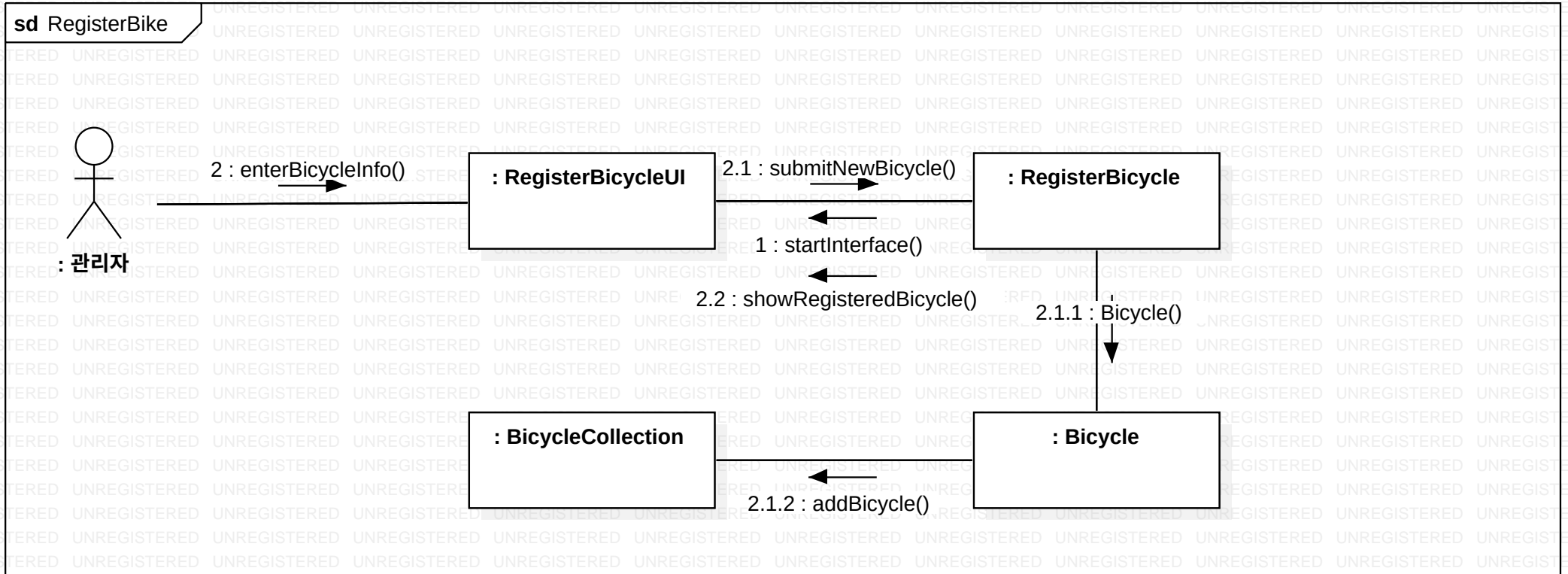
Model::Collaboration5::Interaction1::BicycleRent

sd BicycleRent

: 회원

2 : ☐enterBikeId()

: ☐BicycleRentUI

2.1 : rentBicycle()

1 : startInterface()

2.2 : showRentedBicycle()

: BicycleRent

2.1.1 : findBicycleId()

2.1.2 : rentBicycle()

: BicycleCollection

Model::Collaboration6::Interaction1::□RentedBikeList

sd □RentedBikeList

: 회원

: RentedBicycleListUI

2 : getRentedBicycles()

1 : startInterface()

2.2 : showRentedBicycleList()

: RentedBicycleList

2.1.1 : *[For each bike rented by member]: getRentedBicycles()

: BicycleCollection

Model::ClassDiagram

«boundary»
LogoutUI
-logout_control_: Logout*

«constructor»+LogoutUI(controller: Logout*)
+startInterface(in: istream&, out: ofstream&): void
-showLogoutResult(id: string, out: ofstream&): void

«boundary»
□BicycleRentUI
-bicycle_rent_control_: BicycleRent*

«constructor»+BicycleRentUI(controller: BicycleRent*)
+startInterface(in: istream&, out: ofstream&): void
+enterBicycleId(id: string, out: ofstream&): void
-showRentedBicycle(id: string, product_name: string, out: ofstream&): void

«boundary»
RentedBicycleListUI
-rented_bicycle_list_control_: RentedBicycleList*

«constructor»+RentedBicycleListUI(controller: RentedBicycleList*)
+startInterface(in: istream&, out: ofstream&): void
-showRentedBicycleList(bicycles: vector<const Bicycle*>, out: ofstream&): void

«boundary»
LoginUI
-login_control_: Login*

«constructor»+LoginUI(controller: Signup*)
+startInterface(in: istream&, out: ofstream&): void
+enterCredentials(id: string, password: string, out: ofstream&): void
-showLoginCredential(id: string, password: string, out: ofstream&): void

«control»
Logout
+processLogout(in: istream&, out: ofstream&): void
+processLogoutRequest(): string

«control»
BicycleRent
+processBicycleRent(in: istream&, out: ofstream&): void
+rentBicycle(id: string): Bicycle*

«control»
RentedBicycleList
+processRentedBicycleList(in: istream&, out: ofstream&): void
+getRentedBicycles() const: vector<const Bicycle*>()

«control»
Login
+processLogin(in: istream&, out: ofstream&): void
+validateCredential(id: string, password: string): bool

«boundary»
□SignupUI
-signup_control_: Signup*

«constructor»+SignupUI(controller: Signup*)
+startInterface(in: istream, out: ofstream): void
-enterMemberInfo(id: string, password: string, phone_number: string, out: ofstream): void
-showCreatedMember(id: string, password: string, phone_number: string, out: ofstream): void

«boundary»
RegisterBicycleUI
-register_bicycle_control_: RegisterBicycle*

« constructor»+RegisterBicycleUI(controller: RegisterBicycle*)
+startInterface(in: istream&, out: ofstream&): void
+enterBicycleInfo(id: string, product_name: string, out: ofstream&): void
-showRegisteredBicycle(id: string, product_name: string, out: ofstream&): void

«collection»
MemberCollection
-members_: vector<Member>
-current_member_: Member*

+getInstance(): MemberCollection&
+addMember(member: Member): void
+authenticateMember(id: string, password: string): Member*
+findMemberById(id: string): Member*
+setCurrentMember(member: Member*): void
+getCurrentMember(): Member*
+clearCurrentMember(): void
+isAdminAccount(id: string, password: string): bool
-MemberCollection()

«collection»
BicycleCollection
-bicycles_: vector<Bicycle>

+getInstance(): BicycleCollection&
+addBicycle(bicycle: Bicycle): void
+findBicycleById(id: string): Bicycle*
+getAllBicycles(): const vector<Bicycle>&
-BicycleCollection()

«control»
Signup
+processSignup(in: istream, out: ofstream): void
+submitNewMember(id: string, password: string, phone_number: string): Member

«control»
RegisterBicycle
+processRegisterBicycle(in: istream&, out: ofstream&): void
+submitNewBicycle(id: string, product_name: string): Bicycle

«entity»
Member
-Id
-Password
-PhoneNumber

+Member()
+checkMember()
+getId()
+getPassword()
+getPhoneNumber()

«entity»
Bicycle
-id_: string
-product_name_: string

+Bicycle(id: string, product_name: string)
+getId(): string
+getProductName(): string

+authenticate
+logout
+rent
+requestlist
+register
+register
+rent

1
0..
1
0..
1
0..*