<<control>> LoginSession - loginStates : std::unordered_map<std::string, bool> + setLoginState(id: const std::string&, state: bool) : void + setLogoutState(id: const std::string&) : void isLoggedIn(id: const std::string&) : bool getCurrentLoggedIn(bser() : std::string <<control>> RentalBicycle - bicycleRepository : BicycleRe - reportRental : ReportRental& - loginSession : LoginSession& <<control>> ReportRental - bicycleRepository : BicycleRepository& - loginSession : LoginSession& - rentalRecords : std::vector<RentalInfo> + RentalBicycle(bicycleRepository: BicycleRepository&, reportRental: ReportRental&, loginSession: LoginSession&) + rentalBicycle(bikelD: const std:string&): bool + getBicycleInfo(bikelD: const std::string&): const Bicycle* + ReportRental(bicycleRepository: BicycleRepository&, loginSession: LoginSession(8) + reportBicycle(userID: const std::string8, bikeID: const std::string8, productName: const std::string8) : bool + getRentalInfo(userID: const std::string8) : std::vector<RentalInfo> repo : MemberRepository& validator : ValidateMemberRegistration <<bod><
RentalBicycleUl + Register(repo: MemberRepository&) + RegisterMember(id: const std::string&, pw: const std::string&, phoneNumber: const std::string&): bool rentalBicycle : RentalBicycle& + RentalBicycleUl(rentalBicycle: RentalBicycle&) + startInterface(): void + submitBikeID(): void + showRentBikeID(): const std::string&, productName: const std::string&): void <control>> Logout session : LoginSession& + Logout(session: LoginSession&) + logout(): std::string <
koundary>>
RegisterBicycleUI <<control>> reg : RegisterBicycle& session : LoginSession& reportRental : ReportRental& loginSession : LoginSession& + RegisterBicydeU(reg: RegisterBicycle&, session: LoginSession&) + startInterface(): void + submitBicydeRegistration(): void + showRegisterBicyde(bikeID: const std::string&, productName: const std::string&): void + ReferRental(reportRental: ReportRental&, loginSession: LoginSession&) + referRental() : std::vector<RentalInfo>



<control>>
ValidateMemberRegistration

+ ValidateMemberRegistration(repo: MemberRepository&) + checkFormat(fid: const std::string&) : bool + isDuplicated(fid: const std::string&) : bool + validateRegistration(fid: const std::string&) : bool

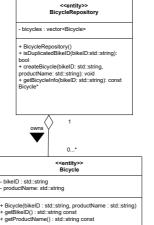
repo : MemberRepository&



-login : Login& + LoginU(login: Login&) + startinterface() : void + submitLogin() : void + submitLogin() : void + showLoginResult(success: bool, id: const std::string&, pw. const std::string&) : void * Story ** Story **

	< keferRentalUl
- referRental : ReferRental&	
+ ReferRentalUI(referRental: F + startInterface(): void + startReferRental(): void + showRentalList(rentalList: co	-,
	< <box> <box> RegisterUl</box></box>

< <boundary>> RegisterUI</boundary>	
reg : Register&	
RegisterUI(reg: Register&) startInterface(): void submitMemberRegistration(): void showeRegister(id: const std::string&, pw: const std::string shoneNumber: const std::string&): void	ı&,





+ Member[id: const std::string&, pw. const std::string&, phoneNumber: const std::string&) + getfd() const std::string&) + getfd() const std::string& + getPhoneNumber(): const std::string& + matchCredentials(inputPw: const std::string&): bool <<fre>fiend>> Login