UML for User part

Zhen Cheng, Qing Lyu

- userId: String + User() + getUserId(): String + setUserId(userId: String): void

PermanentUser

- email: String
- password: String
- userType: String
- friends: ArrayList<String>
- + PermanentUser(email: String, password: String, userType: String)
- + getEmail(): String
- + getPassword(): String
- + getUserType(): String
- + getFriends(): ArrayList<String>
- + setEmail(email: String): void
- + setPassword(password: String): void

UserManager

- permanentUsers: ArrayList<PermanentUser>
- trialUsers: ArrayList<Users>
- suspensions: HashMap<String, Date>
- freezes: ArrayList<String>
- lastLoginTime: HashMap<String, Date>
- tempInfo: HashMap<String, String>
- + UserManager()
- + setPermanentUsers(permanentUsers: ArrayList<PermanentUser>): void
- + getPermanentUsers(): ArrayList<PermanentUser>
- + getFreeze(): ArrayList<String>
- + setFreeze(freezes: ArrayList<String>: void
- + getLastLoginTime(): HashMap<String, Date>
- + setLastLoginTime(lastLoginTime: HashMap<String, Date>): void
- + getSuspensions(): HashMap<String, Date>
- + setSuspensions(HashMap<String, Date>): void
- + getTempInfo(): HashMap<String, String>
- + addFriend(ownEmail: String, friendEmail: String): boolean
- + deleteFriend(ownEmail: String, friendEmail: String): void
- + getFriends(userId: String): ArrayList<String>
- + getFriendsId(userId: String): ArrayList<String>
- + displayUserFriend(userId: String): void
- + displayInfo(email: String): boolean
- + suspendUser(email: String, period: int, suspendTime: long): void
- + checkSuspension(email: String, loginTime: long): boolean
- + displayLeftTime(email: String): Date
- + freezeUser(period: int): void
- + unfreezeUser(): void
- + checkUnfreeze(): ArrayList<String>
- + setLoginTime(email: Stirng, loginTime: long): void
- + userUnfreeze(userId: String): void
- + createUser(email: String, password: String, userType: String): String
- + changeEmail(id: String, newEmail: String): boolean
- + changePwd(id: String, password: String): String
- + pwdType(password: String): String
- + isValidEmail(email: String): boolean
- + checkEmailPwd(email: String, password: String): boolean
- + getUserType(id: String): String
- + getUserTypeByEmail(email: String): String
- + getUserId(email: String): String
- + checkEmail(email: String): boolean

LoginSystem

- um: UserManager
- welMsg: String
- + LoginSystem(um: UserManager)
- + getUm(): UserManager
- + getWelMsg(): String
- + setWelMsg(welMsg: String): void
- + friendAddition(ownEmail: String, friendEmail: String): boolean
- + friendDeletion(ownEmail: String, friendEmail: String): void
- + friendList(email: String): ArrayList<String>
- + displayFriend(email: String): void
- + displayUserInfo(email: String): boolean
- + userCreation(email: String, password: String, userType: String): String
- + userSuspsion(email: String, period: int, suspendTime: long): void
- + userFreeze(period: int): void
- + adminUnfreeze(): void
- + userUnfreeze(email: String): void
- + emailLogin(email: String, password: Stirng): boolean
- + checkSusp(email: String, loginTime: long): boolean
- + setLoginTime(email: String, loginTime: long): void
- + suspTime(email: String): Date
- + emailChange(id: String, email: String): boolean
- + pwdChange(id: String, password: String): String
- + getType(id: String): String
- + getTypeByEmail(email: String): String
- + getPwdType(password: String): String
- + isEMailValid(email: String): boolean
- + forgotPwd(email: String): ArrayList<String
- setTempPwd(userId: String, pwd: String): void
- + checkTemp(email: String, password: String): boolean
- + deleteTemp(email: String): void
- + getId(String email): String
- + emailExist(email: String): boolean