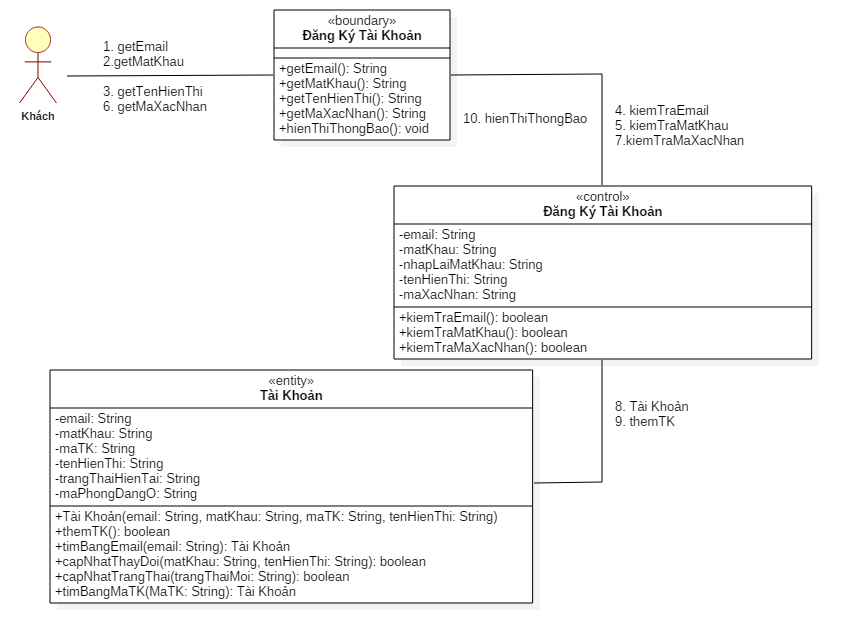
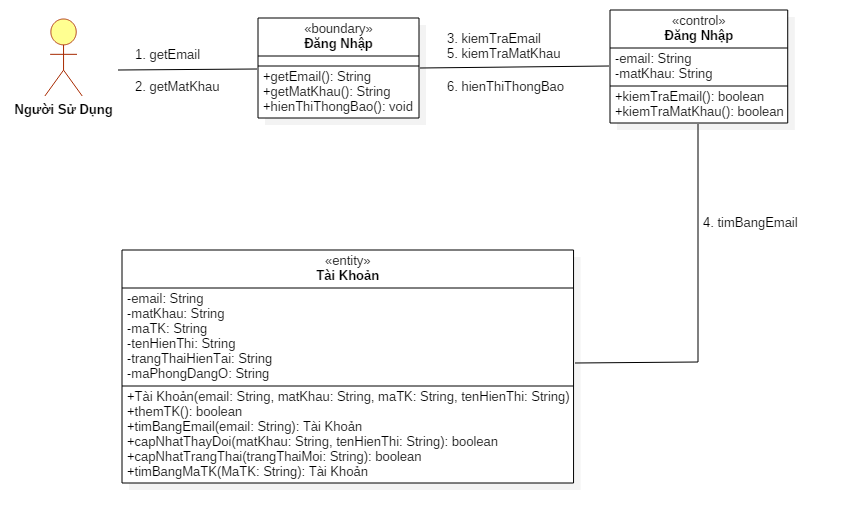
**Báo Cáo Thiết Kế Hệ Thống**

1. **Sơ đồ lớp**
   1. *Quan hệ giữa các lớp trong ca sử dụng Đăng Ký Tài Khoản của Khách*



|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Phương Thức** | **Lớp Chứa** | **Chức Năng** |
| 1 | getEmail() | Đăng Ký Tài Khoản  <<boundary>> | Lấy email được nhập vào từ Khách sau đó chuyển đến lớp control |
| 2 | getMatKhau() | Đăng Ký Tài Khoản  <<boundary>> | Lấy mật khẩu và nhập lại mật khẩu được nhập vào từ Khách sau đó chuyển tới lớp control |
| 3 | getTenHienThi() | Đăng Ký Tài Khoản  <<boundary>> | Lấy tên hiển thị được nhập vào từ Khách sau đó chuyển đến lớp control |
| 4 | kiemTraEmail() | Đăng Ký Tài Khoản  <<control>> | Kiểm tra email vừa được Khách nhập vào có trùng với email đã có trong hệ thống hay không. Kết quả trả về là giá trị boolean: true tương ứng với email đã tồn tại và false tương ứng với trường hợp ngược lại |
| 5 | kiemTraMatKhau() | Đăng Ký Tài Khoản  <<control>> | Kiểm tra xem giá trị mật khẩu và nhập lại mật khẩu được nhập vào bởi Khách có trùng khớp với nhau hay không. Giá trị trả về là true tương ứng với trường hợp là 2 giá trị này trùng khớp và false với trường hợp ngược lại |
| 6 | getMaXacNhan() | Đăng Ký Tài Khoản  <<boundary>> | Lấy mã xác nhận được nhập vàotừ Khách sau đó chuyển tới lớp control. Mã xác nhận là mã mà hệ thống sinh ra rồi gửi vào email đăng ký để xác thực email đăng ký. |
| 7 | kiemTraMaXacNhan() | Đăng Ký Tài Khoản  <<control>> | Kiểm tra xem mã xác nhận vừa được lớp boundary gửi về có chính xác với mã xác nhận đã được sinh ra bởi hệ thống hay không. Kết quả trả về là true tương ứng với trường hợp mã xác nhận này trùng khớp và false tương ứng với trường hợp ngược lại |
| 8 | Tài Khoản( email:String, matKhau:String, maTK:String, tenHienThi:String) | Tài Khoản  <<entity>> | Hàm khởi tạo của lớp Tài Khoản, có chức năng tạo ra 1 đối tượng tài khoản mới, đối tượng này sẽ được gán những thuocj tính tương ứng vói tham số của hàm. Đối tương tạo mới sẽ được tự động gán trangThaiHienTai=“Chưa có chỗ ở” và maPhongDangO=null |
| 9 | themTK() | Tài Khoản  <<entity>> | Thêm tài khoản mà Khách mới đăng ký vào cơ sở dữ liệu. kết quả trả về là true nếu thêm thành công và là false nếu ngược lại. |
| 10 | hienThiThongBao() | Đăng Ký Tài Khoản  <<boundary>> | Đưa ra thông báo cho Khách về kết quả của quá trình đăng ký tài khoản thành công hay thất bại và lý do thất bại (ví dụ như email đã tồn tại, mật khẩu và nhập lại mật khẩu khoogn trùng khớp, mã xác nhận không chính xác…) |

* 1. *Quan hệ giữa các lớp trong ca sử dụng của Người Sử Dụng*
     1. *Quan hệ giữa các lớp trong ca sử dụng Đăng Nhập*



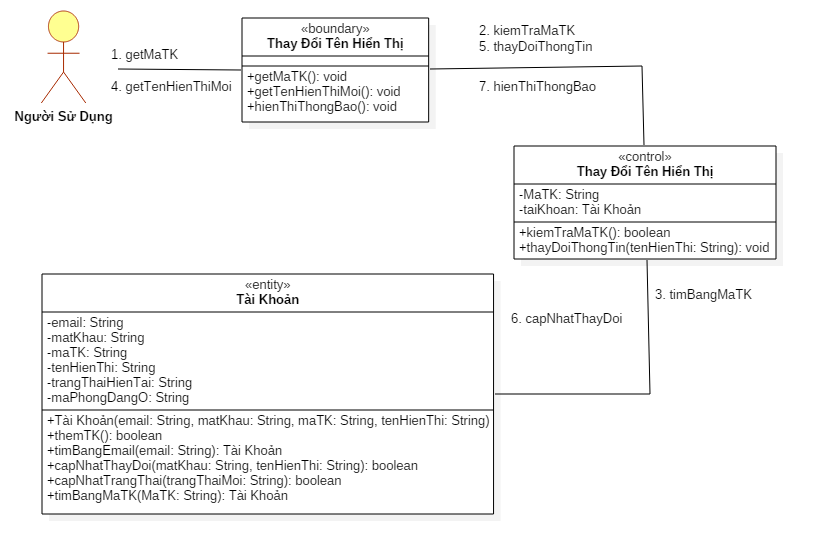
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Phương Thức** | **Lớp Chứa** | **Chức Năng** |
| 1 | getEmail() | Đăng Nhập  <<boundary>> | Lấy email được nhập vào từ Người Sử Dụng rồi chuyển tới lớp control |
| 2 | getMatKhau() | Đăng Nhập  <<boundary>> | Lấy mật khẩu do Người Sử Dụng nhập vào rồi chuyển tới lớp control |
| 3 | kiemTraEmail() | Đăng Nhập  <<control>> | Kiểm tra xem email vừa được nhập vào có tồn tại trong hệ thống hay không bẳng cách lần lượt thực hiện hàm timBangEmail với từng đối tương Tài Khoản. Nếu tồn tại thì kết quả trả về của hàm là true và ngược lại thì kết quả sẽ là false |
| 4 | timBangEmail(email: String) | Tài Khoản  <<entity>> | Tìm kiếm email có tồn tại trong hệ thống hay không bằng cách lần lượt so sánh email với email của các Tài Khoản đã có trong hệ thống. Hàm sẽ làm nhiệm vụ kiểm tra email có trùng với email ở Tài Khoản đang xét hay không.Nếu trùng lặp email, hàm sẽ trả về thông tin Tài Khoản đang xét, ngược lại sẽ trả về null. |
| 5 | kiemTraMatKhau() | Đăng Nhập  <<control>> | Kiểm tra mật khẩu của tài khoản được trả về từ hàm timBangEmail() (nếu tồn tài) với mật khẩu do Người Sử Dụng nhạp vào có trùng khớp hay không. Nếu trùng khớp thì hàm sẽ trả về true, ngược lại thì trả về false |
| 6 | hienThiThongBao() | Đăng Nhập  <<boundary>> | Đưa ra thông báo cho Người Sử Dụng về kết quả của quá trình đăng nhập là thành công hay thất bại và lý do thất bại (ví dụ như email không chính xác hay mật khẩu không chính xác) |

* + 1. *Quan hệ giữa các lớp trong ca sử dụng Tìm Mật Khẩu*



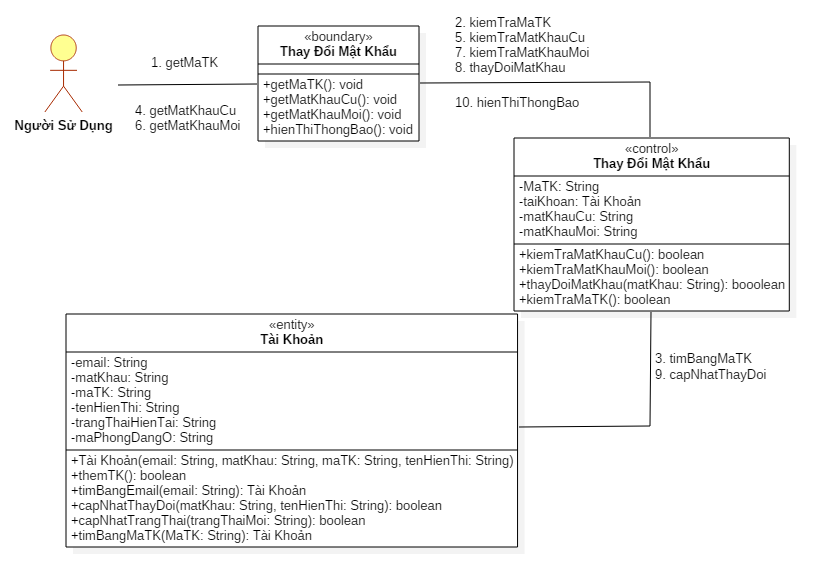
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Phương Thức** | **Lớp Chứa** | **Chức Năng** |
| 1 | getEmail() | Tìm Mật Khẩu  <<boundary>> | Lấy email do Người Sử Dụng nhập vào rồi chuyển đến lóp control |
| 2 | kiemTraEmail() | Tìm Mật Khẩu  <<control>> | Kiểm tra xem email vừa được nhập vào có tồn tại trong hệ thống hay không bẳng cách lần lượt thực hiện hàm timBangEmail với từng đối tương Tài Khoản. Nếu tồn tại thì kết quả trả về của hàm là true và ngược lại thì kết quả sẽ là false |
| 3 | timBangEmail( email: String) | Tài Khoản  <<entity>> | Tìm kiếm email có tồn tại trong hệ thống hay không bằng cách lần lượt so sánh email với email của các Tài Khoản đã có trong hệ thống. Hàm sẽ làm nhiệm vụ kiểm tra email có trùng với email ở Tài Khoản đang xét hay không.Nếu trùng lặp email, hàm sẽ trả về thông tin Tài Khoản đang xét, ngược lại sẽ trả về null. |
| 4 | guiEmail() | Tìm Mật Khẩu  <<control>> | Gửi mật khẩu của tài khoản được trả về từ hàm timBangEmail() (nếu tồn tài) vào email của tài khoản |
| 5 | hienThiThongBao() | Tìm Mật Khẩu  <<boundary>> | Hiển thị thông báo cho Người Sử Dụng là mật khẩu đã gửi về email( trong trường hợp tìm mật khẩu thành công) hoặc đưa ra thông báo rằng email chưa được đăng ký với hệ thống (trong trường họp tìm mật khẩu thất bại) |

* + 1. *Quan hệ giữa các lớp trong ca sử dụng Thay Đổi Thông Tin Cá Nhân*
       1. *Thay Đổi Tên Hiển Thị*



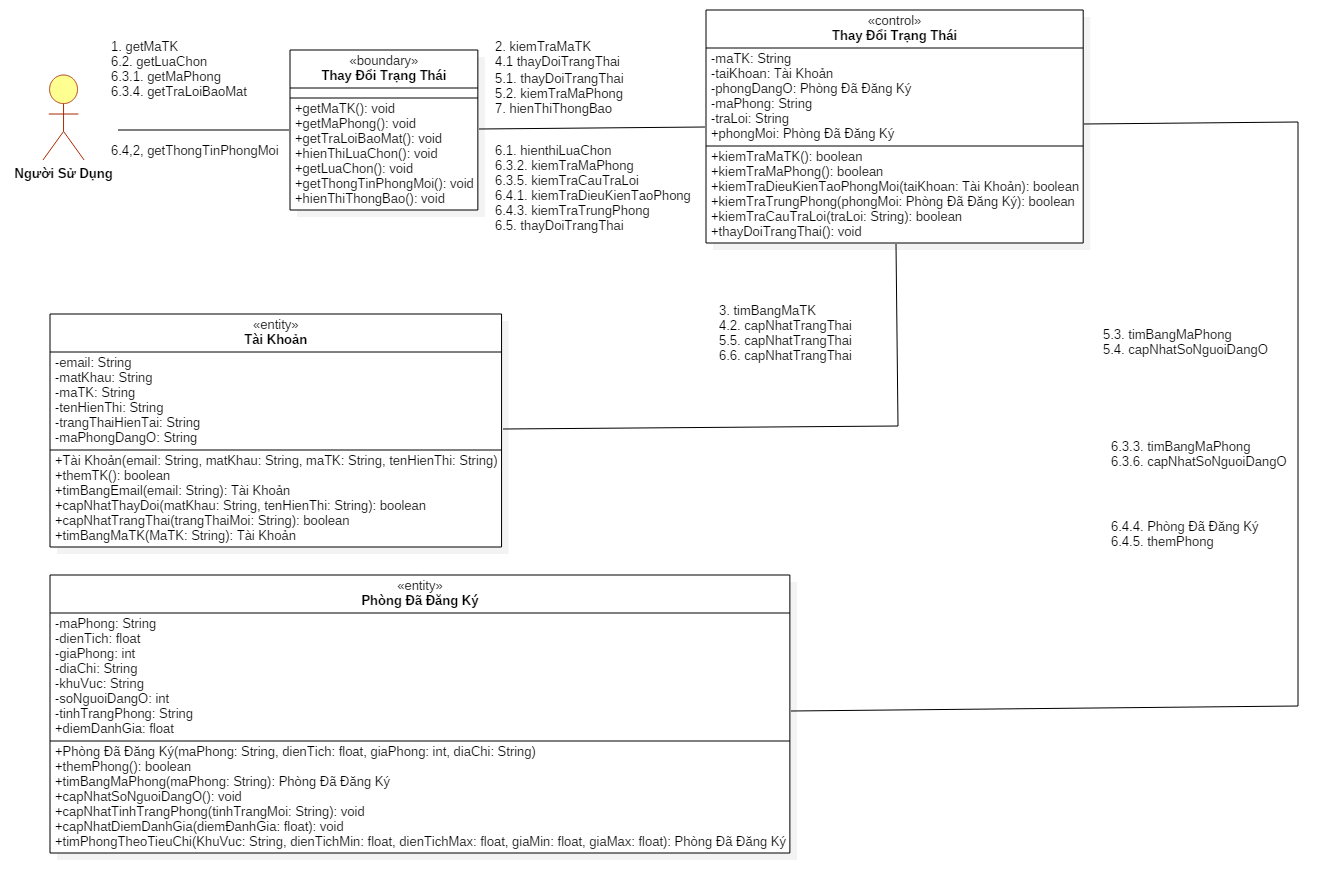
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Phương Thức** | **Lớp Chứa** | **Chức Năng** |
| 1 | getMaTK() | Thay Đổi Tên Hiển Thị  <<boundary>> | Lấy mã tài khoản từ tài khoản mà người sử dụng đã đăng nhập vào hệ thống chuyển đến lớp control |
| 2 | kiemTraMaTK() | Thay Đổi Tên Hiển Thị  <<control>> | Kiểm tra xem tài khoản tương ứng với mã tài khoản có tồn tại trong hệ thống hay không bằng cách thực hiện lần lượt hàm timBangMaTK của từng đối tượng Tài Khoản. Kết quả trả về là true nếu tồn tại và false nếu ngược lại |
| 3 | timBangMaTK(maTK: String) | Tài Khoản  <<entity>> | So sánh mã tài khoản mà lớp boundary gửi tới cho lớp control với mã tài khoản của đối tượng Tài Khoản đang xét trong CSDL. Kết quả trả về sẽ là Tài Khoản đang xét nếu 2 mã tài khoản là trùng nhau và trả về null nếu ngươc lại. Hệ thống sẽ tiến hành duyệt hàm này của từng đối tượng Tài Khoản cho đến khi gặp đối tượng Tài Khoản cần tìm. |
| 4 | getTenHienThiMoi() | Thay đổi Tên Hiển Thị  <<boundary>> | Lấy tên hiển thị mới do người dùng nhập vào rồi chuyển tới lớp control |
| 5 | thayDoiThongTin( tenHienThi: String) | Thay đổi Tên Hiển Thị  <<control>> | Tiến hành thay đổi tên hiển thị và lưu trữ tạm thời thay đổi này để chuyển tới lớp entity |
| 6 | capNhatThayDoi(matKhau: String, tenHienThi: String) | Tài Khoản  <<entity>> | Cập nhật tên hiển thị và mật khẩu mới cho Người Sử Dụng vào cơ sở dữ liệu |
| 7 | hienThiThongBao() | Thay Đổi Tên Hiển Thị  <<boundary>> | Hiển thị thông báo cho Người Sử Dụng rằng thay đổi tên hiển thị thành công |

* + - 1. *Thay Đổi Mật Khẩu*



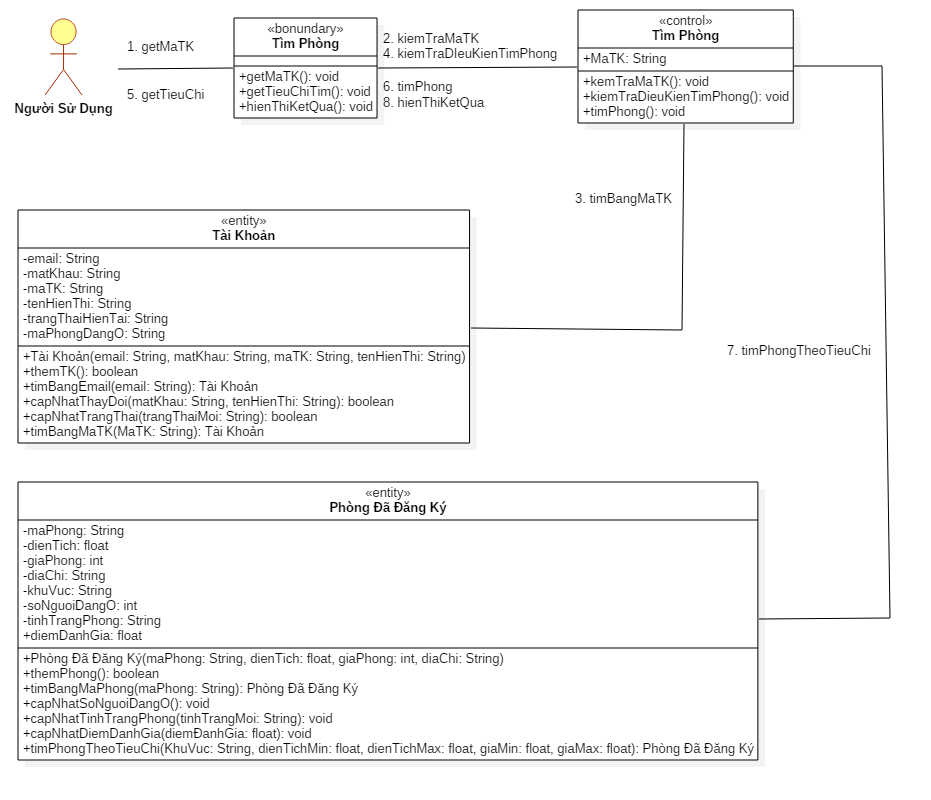
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Phương Thức** | **Lớp Chứa** | **Chức Năng** |
| 1 | getMaTK() | Thay Đổi Mật Khẩu  <<boundary>> | Lấy mã tài khoản từ tài khoản mà người sử dụng đã đăng nhập vào hệ thống chuyển đến lớp control |
| 2 | kiemTraMaTK() | Thay Đổi Mật Khẩu  <<control>> | Kiểm tra xem tài khoản tương ứng với mã tài khoản có tồn tại trong hệ thống hay không bằng cách thực hiện lần lượt hàm timBangMaTK của từng đối tượng Tài Khoản. Kết quả trả về là true nếu tồn tại và false nếu ngược lại |
| 3 | timBangMaTK(maTK: String) | Tài Khoản  <<entity>> | So sánh mã tài khoản mà lớp boundary gửi tới cho lớp control với mã tài khoản của đối tượng Tài Khoản đang xét trong CSDL. Kết quả trả về sẽ là Tài Khoản đang xét nếu 2 mã tài khoản là trùng nhau và trả về null nếu ngươc lại. Hệ thống sẽ tiến hành duyệt hàm này của từng đối tượng Tài Khoản cho đến khi gặp đối tượng Tài Khoản cần tìm. |
| 4 | getMatKhauCu() | Thay Đổi Mật Khẩu  <<boundary>> | Lấy mật khẩu cũ do Người Sử Dụng nhập vào rồi chuyển tới lớp control |
| 5 | kiemTraMatKhauCu() | Thay Đổi Mật Khẩu  <<control>> | Kiểm tra mật khẩu cũ mà Người Sử Dụng đã nhập có trùng khớp với mật khẩu của thực thể đối tượng được trả về bởi hàm timBangMaTK hay không. Nếu trùng khớp thì trả về true và ngược lại sẽ trả về false |
| 6 | getMatKhauMoi() | Thay Đổi Mật Khẩu  <<boundary>> | Lấy mật khẩu mới và nhập lại mật khẩu do Người Sử Dụng nhập vào rồi chuyển cho lớp control |
| 7 | kiemTraMatKhauMoi() | Thay Đổi Mật Khẩu  <<control>> | Kiểm tra xem mật khuẩ mới và nhập lại mật khẩu do lớp boundary chuyển về có trùng khớp hay không. Kết quả trả về là true nếu trùng khớp và false nếu ngược lại. |
| 8 | thayDoiMatKhau(matKhau: String) | Thay Đổi Mật Khẩu  <<control>> | Tiến hành thay đổi mật khẩu và lưu trữ tạm thời thay đổi này để chuyển tới lớp entity |
| 9 | capNhatThayDoi(matKhau: String, tenHienThi: String) | Tài Khoản  <<entity>> | Cập nhật tên hiển thị và mật khẩu mới cho Người Sử Dụng vào cơ sở dữ liệu |
| 10 | hienThiThongBao() | Thay Đổi Mật Khẩu  <<boundary>> | Hiển thị thông báo cho Người Sử Dụng biết kết quả của quá trình thay đổi mật khẩu là thành công hay thất bại và lý do thất bại ( mật khẩu cũ không chính xác, mật khẩu mới và nhập lại mật khẩu không trùng khớp) |

* + 1. *Quan hệ giữa các lớp trong ca sử dụng Thay Đổi Trạng Thái Tài Khoản*



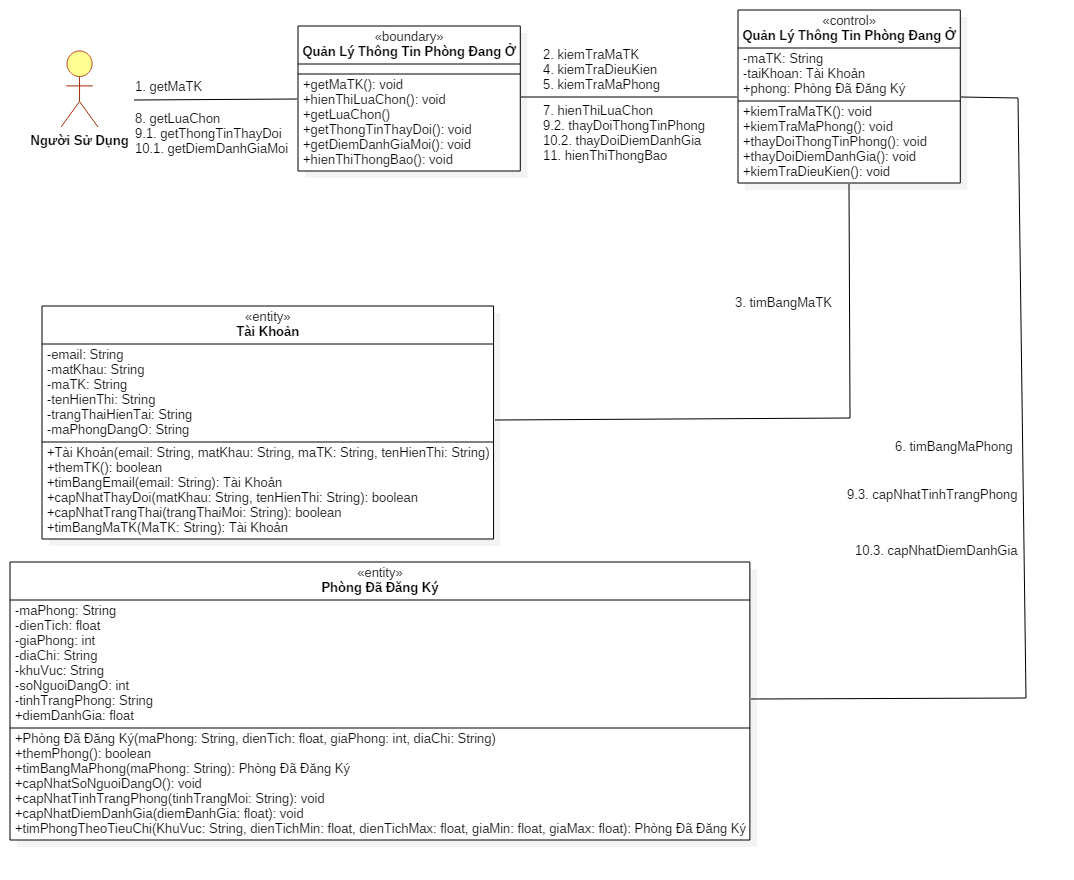
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Phương Thức** | **Lớp Chứa** | **Chức Năng** |
| 1 | getMaTK() | Thay Đổi Trạng Thái  <<boundary>> | Lấy mã tài khoản từ tài khoản mà người sử dụng đã đăng nhập vào hệ thống chuyển đến lớp control |
| 2 | kiemTraMaTK() | Thay Đổi Trạng Thái  <<control>> | Kiểm tra xem tài khoản tương ứng với mã tài khoản có tồn tại trong hệ thống hay không bằng cách thực hiện lần lượt hàm timBangMaTK |
| 3 | timBangMaTK(maTK: String) | Tài Khoản  <<entity>> | So sánh mã tài khoản mà lớp boundary gửi tới cho lớp control với mã tài khoản của đối tượng Tài Khoản đang xét trong CSDL. Kết quả trả về sẽ là Tài Khoản đang xét nếu 2 mã tài khoản là trùng nhau và trả về null nếu ngươc lại. Hệ thống sẽ tiến hành duyệt hàm này của từng đối tượng Tài Khoản cho đến khi gặp đối tượng Tài Khoản cần tìm. |
| 4 | **Nếu trạng thái của tài khoản trả về là “Đã có chỗ ở”** | | |
| 4..1 | thayDoiTrangThai() | Thay Đổi Trạng Thái  <<control>> | Tiến hành thay đổi trạng thái của Tài Khoản của Người Sử Dụng yêu cầu từ “Đã có chỗ ở” sang “muốn chuyển” và chuyển thay đổi này tới lớp entity |
| 4.2 | capNhatTrangThai( trangThaiMoi: String) | Tài Khoản  <<entity>> | Cập nhật trạng thái mới cho Người Sử Dụng vào cơ sở dữ liệu |
| 5 | **Nếu trạng thái của tài khoản trả về là “Muốn chuyển”** | | |
| 5.1 | thayDoiTrangThai() | Thay Đổi Trạng Thái  <<control>> | Tiến hành thay đổi trạng thái của Tài Khoản của Người Sử Dụng yêu cầu từ “Muốn chuyển” sang “Chưa có chỗ ở” và chuyển thay đổi này tới lớp Tài Khoản <<entity>> |
| 5.2 | kiemTraMaPhong() | Thay Đổi Trạng Thái  <<control>> | Kiểm tra xem Phòng Đã Đăng Ký tương ứng với mã phòng có tồn tại hay không băng cách thực hiện lần lượt hàm timBangMaPhong của từng đối tượng Phòng Đã Đăng Ký. Kết quả trả về là true nếu tồn tại và false nếu ngược lại |
| 5.3 | timBangMaPhong( maPhong: String) | Phòng Đã Đăng Ký  <<entity>> | So sánh mã phòng ở trường maPhongDangO của đối tượng Tài Khoản đang xét với mã phòng của 1 thực thể Phòng Đã Đăng Ký. Nếu trùng nhau thì sẽ trả về thực thể Phòng Đã Đăng Ký đang xét, ngược lại trả về null. Hệ thống sẽ tiến hành duyệt lần lượt từng thực thể phòng cho tói khi gặp thực thể cần tìm. |
| 5.4 | capNhatSoNguoiDangO() | Phòng Đã Đăng Ký  <<entity>> | Cập nhật lại số người đang ở trong phòng sau khi 1 Tài Khoàn đã rời phòng, Nếu số người trong phòng bằng 0 thì thiết lập trạng thai s của phòng thành “Trống” |
| 5.5 | capNhatTrangThai( trangThaiMoi: String) | Tài Khoản  <<entity>> | Cập nhật trạng thái mới đồng thời cập nhật lại mã phòng đang ở thành null cho Người Sử Dụng vào cơ sở dữ liệu |
| 6 | **Nếu trạng thái của tài khoản trả về là “Chưa có chỗ ở”** | | |
| 6.1 | hienThiLuaChon() | Thay Đổi Trạng Thái  <<boundary>> | Hiển thị lựa chọn để Người Sử Dụng tiến hành lựa chọn, bao gồm: Ở phòng trong hệ thống hay ở phòng mới |
| 6.2 | getLuaChon() | Thay Đổi Trạng Thái  <<boundary>> | Lấy lựa chọn của Người Sử Dụng và chuyển đến lớp control |
| 6.3 | **Nếu Người Sử Dụng lựa chọn Ở phòng đã có trong hệ thống** | | |
| 6.3.1 | getMaPhong() | Thay Đổi Trạng Thái  <<boundary>> | Lấy mã phòng mà Người Sử Dụng đã điền vào và chuyển tới lớp control |
| 6.3.2 | kiemTraMaPhong() | Thay Đổi Trang Thái  <<control>> | Kiểm tra xem Phòng Đã Đăng Ký tương ứng với mã phòng có tồn tại hay không băng cách thực hiện lần lượt hàm timBangMaPhong của từng đối tượng Phòng Đã Đăng Ký. Kết quả trả về là true nếu tồn tại và false nếu ngược lại |
| 6.3.3 | timBangMaPhong( MaPhong: String) | Phòng Đã Đăng Ký  <<entity>> | So sánh mã phòng mà Người Sử Dụng nhập vào với mã phòng của 1 thực thể Phòng Đã Đăng Ký. Nếu trùng nhau thì sẽ trả về thực thể Phòng Đã Đăng Ký đang xét, ngược lại trả về null. Hệ thống sẽ tiến hành duyệt lần lượt từng thực thể phòng cho tới khi gặp thực thể cần tìm hoặc hết tất cả các đối tượng Phòng Đã Đăng Ký. |
| 6.3.4 | getTraLoiBaoMat() | Thay Đổi Trạng Thái  <<boundary>> | Lấy câu trả lời của câu hỏi bảo mật phòng tử Người Sử Dụng và chuyển tới lớp control |
| 6.3.5 | kiemTraCauTraLoi(traLoi: String) | Thay Đổi Trạng Thái  <<control>> | Kiểm tra xem câu trả lời của Người Sử dụng có chính xác hay không. Kết quả trả về là true nếu chính xác và false nếu ngược lại |
| 6.3.6 | capNhatSo NguoiDangO() | Phòng Đã Đăng Ký  <<entity>> | Cập nhật lại số tài khoản đang ở trong phòng sau khi có 1 tài khoản mới chuyển vào phòng |
| 6.4 | **Nếu Người Sử Dụng lựa chọn Tạo phòng mới** | | |
| 6.4.1 | kiemTraDieuKienTaoPhong( taiKhoan: Tài Khoản) | Thay Đổi Trạng Thái  <<control>> | Kiểm tra điều kiện tạo phòng mới của tài khoản: mỗi tài khoản chỉ được tạo phòng mới duy nhất 1 lần |
| 6.4.2 | getThongTinPhongMoi() | Thay Đổi Trạng Thái  <<boundary>> | Lấy các thông tin về phòng mới do Người Sử Dụng nhập vào bao gồm: địa điểm, khu vực, diện tích, giá cả, các thông tin khác và chuyển tới lớp control |
| 6.4.3 | kiemTraPhongTrung( phongMoi: Phòng Đã Đăng Ký) | Thay Đổi Trạng Thái  <<control>> | Kiểm tra xem các thông tin về phòng mới do Người Sử Dụng nhập vào có trùng lặp với phòng nào đã có trong hệ thống hay không. Kết quả trả về là true nếu có trùng lặp và false nếu ngược lại |
| 6.4.4 | Phòng Đã Đăng Ký(maPhong: String, dienTich: float, giaPhong: int, diaChi: String) | Phòng Đã Đăng Ký  <<entity>> | Đây là hàm khởi tạo của lớp Phòng Đã Đăng Ký ở với chức năng khởi tạo 1 đối tượng Phòng Đã Đăng Ký mới. |
| 6.4.5 | themPhong() | Phòng Đã Đăng Ký  <<entity>> | Thêm Phòng Đã Đăng Ký vừa khởi tạo vào cơ sở dữ liệu. |
| 6.5 | thayDoiTrangThai() | Thay Đổi Trạng Thái  <<control>> | Tiến hành thay đổi trạng thái của Tài Khoản của Người Sử Dụng yêu cầu từ “Chưa có chỗ ở” sang “Đã có chỗ ở” và chuyển thay đổi này tới lớp Tài Khoản <<entity>> |
| 6.6 | capNhatTrangThai( trangThaiMoi: String) | Tài Khoản  <<entity>> | Cập nhật trạng thái mới đồng thời cập nhật lại mã phòng đang ở cho Người Sử Dụng vào cơ sở dữ liệu |
| 7 | hienThiThongBao() | Thay Đổi Trạng Thái <<boundary>> | Hiển thị thông báo cho Người Sử Dụng về kết quả của quá trình thay đổi trạng thái |

* + 1. *Quan hệ giữa các lớp trong ca sử dụng Tìm Phòng*



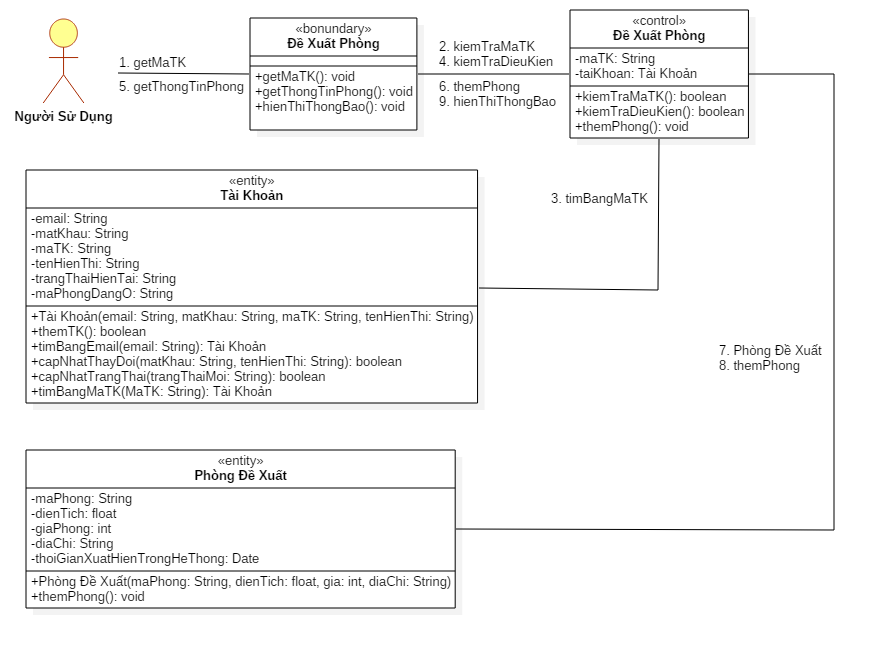
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Phương Thức** | **Lớp Chứa** | **Chức Năng** |
| 1 | getMaTK() | Tìm Phòng  <<boundary>> | Lấy mã tài khoản từ tài khoản mà người sử dụng đã đăng nhập vào hệ thống chuyển đến lớp control |
| 2 | kiemTraMaTK() | Tìm Phòng  <<boundary>> | Kiểm tra xem tài khoản tương ứng với mã tài khoản có tồn tại trong hệ thống hay không bằng cách thực hiện lần lượt hàm timBangMaTK của từng đối tượng Tài Khoản. Kết quả trả về là true nếu tồn tại và false nếu ngược lại |
| 3 | timBangMaTK( maTK: String) | Tài Khoản  <<entity>> | So sánh mã tài khoản mà lớp boundary gửi tới cho lớp control với mã tài khoản của đối tượng Tài Khoản đang xét trong CSDL. Kết quả trả về sẽ là Tài Khoản đang xét nếu 2 mã tài khoản là trùng nhau và trả về null nếu ngươc lại. Hệ thống sẽ tiến hành duyệt hàm này của từng đối tượng Tài Khoản cho đến khi gặp đối tượng Tài Khoản cần tìm. |
| 4 | kiemTraDieuKienTimPhong() | Tìm Phòng  <<control>> | Kiểm tra điều kiện tìm phòng của tài khoản: chỉ tài khoản có trạng thái “Chưa có chỗ ở” hoặc “Muốn tìm” mới có chức năng tìm phòng |
| 5 | getTieuChi() | Tìm Phòng  <<entity>> | Lấy các tiêu chí về phòng cần tìm do Người Sử Dụng nhập vào và chuyển tới lớp control |
| 6 | timPhong() | Tìm Phòng  <<control>> | Tìm phòng thỏa mãn tiêu chí của Người Sử Dụng bằng cách thực hiện lần lượt các hàm timPhongTheoTieuChi của các đối tượng Phòng Đã Đăng Ký |
| 7 | timPhongTheoTieuChi( khuVuc: String, dienTichMin: float, dienTichMax: float, giaMin: float, giaMax: float) | Phòng Đã Đăng Ký  <<entity>> | Thực hiện kiểm tra xem đối tượng PHòng Đã Đăng Ký đang xét có thỏa mãn các tiêu chí hay không. Kết quả trả về là đối tượng đang xét nếu thỏa mãn và null nếu ngược lại |
| 8 | hienThiKetQua() | Tìm Phòng  <<boundary>> | Hiển thị ra kết quả tìm kiếm là danh sách các phòng thỏa mãn được sắp xếp theo thứ tự giảm dần của điểm đánh giá |

* + 1. *Quan hệ giữa các lớp trong ca sử dụng Quản Lý Thông Tin Phòng Đang Ở*



|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Phương Thức** | **Lớp Chứa** | **Chức Năng** |
| 1 | getMaTK() | Quản Lý Thông Tin Phòng Đang Ở  <<boundary>> | Lấy mã tài khoản từ tài khoản mà người sử dụng đã đăng nhập vào hệ thống chuyển đến lớp control |
| 2 | kiemTraMaTK() | Quản Lý Thông Tin Phòng Đang Ở  <<control>> | Kiểm tra xem tài khoản tương ứng với mã tài khoản có tồn tại trong hệ thống hay không bằng cách thực hiện lần lượt hàm timBangMaTK của từng đối tượng Tài Khoản. |
| 3 | timBangMaTK( maTK: String) | Tài Khoản  <<entity>> | So sánh mã tài khoản mà lớp boundary gửi tới cho lớp control với mã tài khoản của đối tượng Tài Khoản đang xét trong CSDL. Kết quả trả về sẽ là Tài Khoản đang xét nếu 2 mã tài khoản là trùng nhau và trả về null nếu ngươc lại. Hệ thống sẽ tiến hành duyệt hàm này của từng đối tượng Tài Khoản cho đến khi gặp đối tượng Tài Khoản cần tìm. |
| 4 | kiemTraDieuKien() | Quản Lý Thông Tin Phòng Đang Ở  <<control>> | Kiểm tra xem đối tượng Tài Khoản đang xét có thỏa mãn điều kiện quản lý thông tin phòng đang ở hay không: chỉ có tài khoản có trạng thái “Đã có chỗ ở” hoặc “Muốn chuyền” mới có khả năng quản lý trạng thái phòng đang ở. |
| 5 | kiemTraMaPhong() | Quản Lý Thông Tin Phòng Đang Ở  <<control>> | Kiểm tra xem Phòng Đã Đăng Ký tương ứng với mã phòng có tồn tại hay không băng cách thực hiện lần lượt hàm timBangMaPhong của từng đối tượng Phòng Đã Đăng Ký. |
| 6. | timBangMaPhong( maPhong: String) | Phòng Đã Đăng Ký  <<entity>> | So sánh mã phòng ở trường maPhongDangO của đối tượng Tài Khoản đang xét với mã phòng của 1 thực thể Phòng Đã Đăng Ký. Nếu trùng nhau thì sẽ trả về thực thể Phòng Đã Đăng Ký đang xét, ngược lại trả về null. Hệ thống sẽ tiến hành duyệt lần lượt từng thực thể phòng cho tói khi gặp thực thể cần tìm. |
| 7 | hienThiLuaChon() | Quản Lý Thông Tin Phòng Đang Ở  <<boundary>> | Đưa ra lựa chọn cho Người Sử Dụng bao gồm 2 lựa chọn: thay đổi điểm đánh giá hay thay đổi thông tin khác. |
| 8 | getLuaChon() | Quản Lý Thông Tin Phòng Đang Ở  <<boundary>> | Lấy lựa chọn của người sử dụng chuyển tới lớp control để đưa các xử lý tiếp theo |
| 9 | **Nếu Người Sử Dụng lựa chọn thay đổi các thông tin khác** | | |
| 9.1 | getThongTinThayDoi() | Quản Lý Thông Tin Phòng Đang Ở  <<boundary>> | Lấy các thông tin mà Người Sử dụng muốn thay đổi về phòng rồi chuyển tới lớp control |
| 9.2 | thayDoiThongTinPhong() | Quản Lý Thông Tin Phòng Đang Ở  <<control>> | Thực hiện thay đổi các thông tin của đối tượng phòng đang xét theo yêu cầu của ngươì sử dụng và chuyển các thay đổi này tới lớp Phòng Đã Đăng Ký <<entity>> |
| 9.3 | capNhatTinhTrangPhong() | Phòng Đã Đăng Ký  <<entity>> | Tiến hành cập nhật các thông tin thay đổi mà người sử dụng mong muốn vào cơ sở dữ liệu |
| 10 | **Nếu Người Sử Dụng lụa chọn thay đổi điểm đánh giá** | | |
| 10.1 | getDiemDanhGiaMoi() | Quản Lý Thông Tin Phòng Đang Ở  <<boundary>> | Lấy điểm đánh giá mới về phòng từ người sử dụng và chuyển tới lớp control |
| 10.2 | thayDoiDiemDanhGIa() | Quản Lý Thông Tin Phòng Đang Ở  <<control>> | Tính toán lại điểm đánh giá của đối tượng phòng đang xét dựa trên điểm đánh giá do người sử dụng mới thay đổi và chuyển điểm này tới lớp Phòng Đã Đăng Ký <<entity>> |
| 10.3 | capNhatDiemDanhGia( diemDanhGia: float) | Phòng Đã Đăng Ký  <<entity>> | Cập nhật điểm đánh giá mới do hệ thống tính toán của đối tương phòng đang xét vào cơ sở dữ liệu |
| 11 | hienThiThongBao() | Quản Lý Thông Tin Phòng Đang Ở  <<boundary>> | Hiển thị thoogn báo cho Người Sử Dụng về kết quả của quá trình thay đổi thông tin phòng đang ở. |

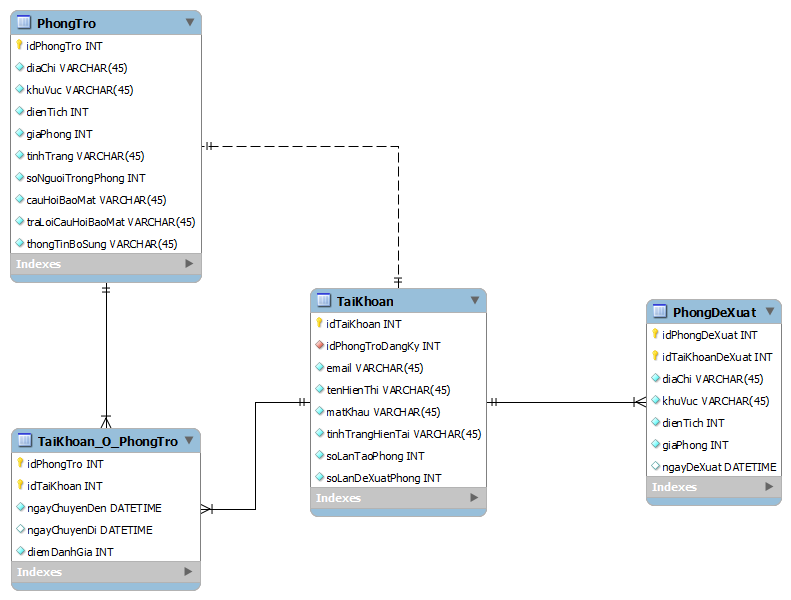
* + 1. *Quan hệ giữa các lớp trong ca sử dụng Đề Xuất Phòng*



|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Phương Thức** | **Lớp Chứa** | **Chức năng** |
| 1 | getMaTK() | Đề Xuất Phòng  <<boundary>> | Lấy mã tài khoản từ tài khoản mà người sử dụng đã đăng nhập vào hệ thống chuyển đến lớp control |
| 2 | kiemTraMaTK() | Đề Xuất Phòng  <<control>> | Kiểm tra xem tài khoản tương ứng với mã tài khoản có tồn tại trong hệ thống hay không bằng cách thực hiện lần lượt hàm timBangMaTK của từng đối tượng Tài Khoản. Kết quả trả về là true nếu tồn tại và false nếu ngược lại |
| 3 | timBangMaTK( maTK: String) | Tài Khoản  <<entity>> | So sánh mã tài khoản mà lớp boundary gửi tới cho lớp control với mã tài khoản của đối tượng Tài Khoản đang xét trong CSDL. Kết quả trả về sẽ là Tài Khoản đang xét nếu 2 mã tài khoản là trùng nhau và trả về null nếu ngươc lại. Hệ thống sẽ tiến hành duyệt hàm này của từng đối tượng Tài Khoản cho đến khi gặp đối tượng Tài Khoản cần tìm. |
| 4 | kiemTraDieuKien() | Đề Xuất Phòng  <<control>> | Kiểm tra xem taiKhoan đang xét có thỏa mãn điều kiện đề xuất phòng hay không: chỉ có tài khoản có trạng thái “Đã có chỗ ở” hoặc “Muốn chuyển” đồng thời số lần đề xuất phòng chưa quá 2 lần mới có khả năng đề xuất phòng |
| 5 | getThongTinPhong() | Đề Xuất Phòng  <<boundary>> | Lấy các thông tin về phòng mà người sử dụng muốn đề xuất do chính họ nhập vào cà chuyển tới lớp control để xử lý |
| 6 | themPhong() | Đề Xuất Phòng  <<control>> | Tiến hành tiếp nhận các thông tin được chuyển từ lớp boundary và sẽ tạo ra 1 đối tương phòng đề xuất mới đồng thời thêm đối tương này vào cơ sở dữ liệu |
| 7 | Phòng Đề Xuất( maPhong: String, dienTich: float, gia:int, diaChi: String) | Phòng Đề Xuất  <<entity>> | Là hàm khởi tạo của lớp Phòng Đề Xuất <<entity>>, có chức năng khởi tạo một đối tượng Phòng Đề Xuất mới có các thuộc tính chính bằng giá trị các tham số truyền vào, đồng thời thuộc tính thoiGianXuatHienTrongHeTHong sẽ được gán chính bằng thời gian hệ thống khởi tạo đói tượng |
| 8 | themPhong() | Phòng Đề Xuất  <<entity>> | Thêm đối tương phòng đề xuất vừa tạo vào cơ sở dữ liệu |
| 9 | hienThiThongBao() | Đề Xuất Phòng  <<boundary>> | Hiển thị thông báo cho Người Sử Dụng về kết quả của quá trình đề xuất phòng là thành công hay thất bại và lý do thất bại(Tài khoản không thỏa mãn điều kiện đề xuất phòng) |

1. **Thiết kế cơ sở dữ liệu**

* *Mô hình thực thể liên kết ERD*

****

*Chú thích:*

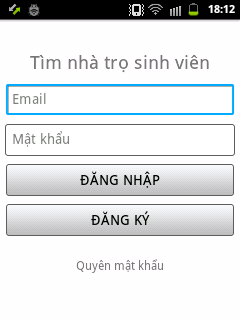
Trong sơ đồ thực thể liên kết gồm 3 thực thể: **PhongTro**, **TaiKhoan**,**PhongDeXuat**. Các liên kết bao gồm:

+) **TaiKhoan\_O\_PhongTro**: là liên kết nhiều-nhiều với mỗi liên kết cần lưu trữ: *ngayChuyenDen*, *ngayChuyenDi* và *diemDanhGia.*

+) **TaiKhoan\_Tao\_PhongTro**: Là liên kết 1-1,0 thể hiện một tài khoản chỉ tạo được tối đa một phòng trọ.

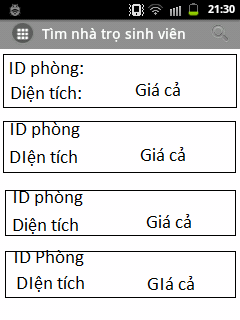
+) **TaiKhoan\_Tao\_PhongDeXuat**: là liên kết 1-0,2 thể hiện một tài khoản có thể tạo tối đa 2 phòng đề xuất.

1. **Hướng thiết kế giao diện**
   1. *Giao diện đăng nhập*



Tại giao diện này cung cấp tính năng đăng nhập, đăng ký, tìm mật khẩu

* 1. *Khi người dùng chọn vào đăng nhập, nếu thành công thì sẽ vào giao diện chính của ứng dụng*



Trong phần giao diện chính sẽ hiển thị một số vài thông tin cơ bản của các phòng trong hệ thống bao gồm cả phòng đề xuất và phòng đăng ký, khi người dùng chọn vào một item trên danh sách thì sẽ vào giao diện thông tin chi tiết của phòng được chọn.

Giao diện này cung cấp tính năng tìm phòng (Khi người dùng chọn vào nút search ở góc trên bên phải màn hình,

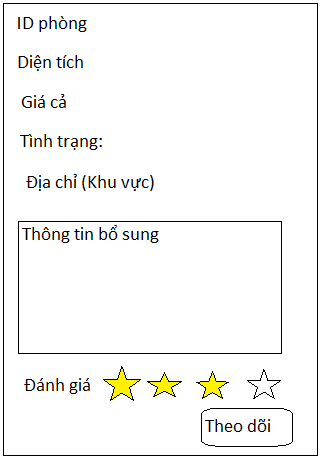
Góc trên bên trái có nút menu vào menu chính của ứng dụng

* 1. *Giao diện menu:*



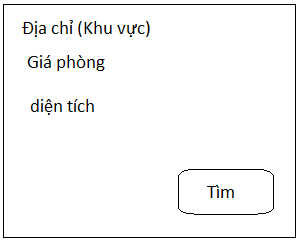
Trong menu người dùng có thể chọn các chức năng tương ứng

1. Phòng mới sẽ hiển thị những phòng mới xuất hiện trên hệ thống(cả phòng đề xuất + phòng đăng ký) mặc định khi vào giao diện chính sẽ hiển thị thông tin này.
2. Đang theo dõi hiển thị những phòng mà người dùng quan tâm (Trong quá trình tìm phòng) người dùng có thể theo dõi những phòng ưng ý)
3. Thông tin các nhân: vào chức năng thay đổi thông tin cá nhân
4. Quản lý phòng: vào chức năng quản lý phòng
5. Đăng ký phòng: vào chức năng tạo phòng
6. Đề xuất phòng: vào chức năng đề xuất phòng
   1. *Giao diện chi tiết của phòng*



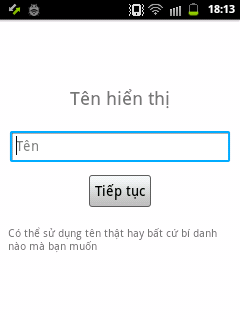
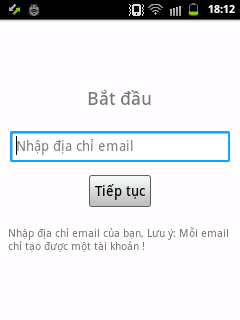
Giao diện cung cấp thông tin về phòng mà người dùng chọn, Có nút bấm theo dõi để người dùng có thể tiện tìm lại trong những lần sau.

* 1. *Giao diện tìm phòng*

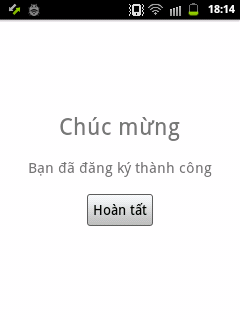


Kết quả tìm kiếm trả về thành danh sách các phòng đăng ký và đề xuất

* 1. *Giao diện đăng ký tài khoản:*

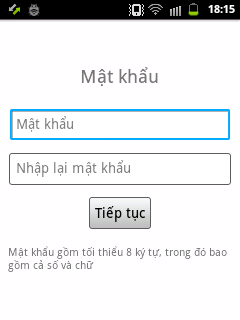




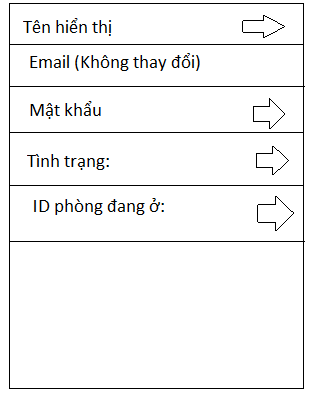


* 1. *Giao diện tìm mật khẩu:*





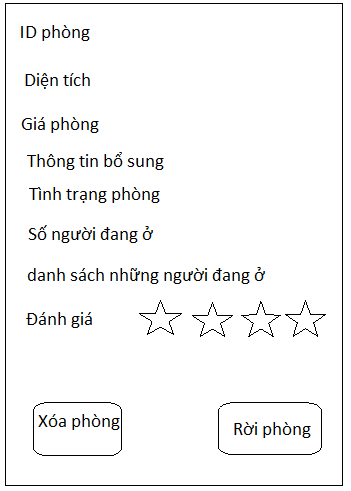
* 1. *Thay đổi thông tin cá nhân*



Cung cấp tính năng thay đổi thông tin như tên hiển thị, mật khẩu, tình trạng, Id phòng đang ở

(Khi thay đổi id phòng đang ở người dùng cũng phải trả lời đúng câu hỏi bảo mật tương ứng)

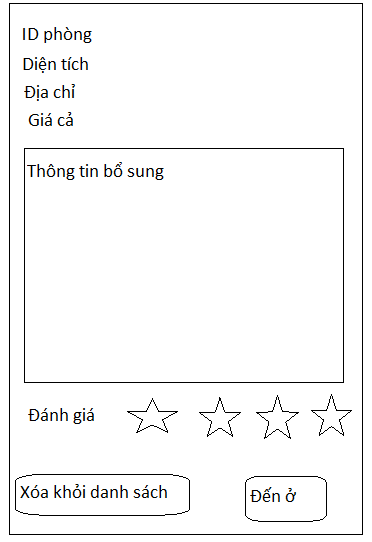
* 1. *Quản lý thông tin phòng*



Người dùng có thể thay đổi các thông tin phòng trừ Id phòng.

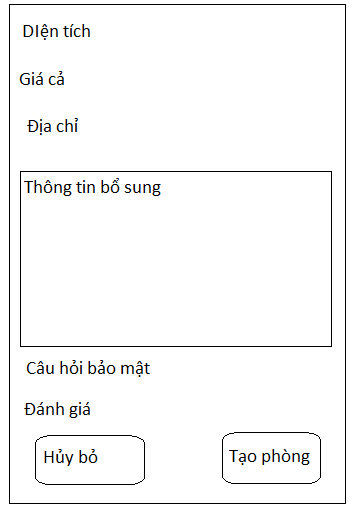
* 1. *Đến ở phòng trong hệ thống*

Khi người dùng tìm thấy phòng bằng công cụ tìm kiếm chọn vào nút Theo dõi, Phòng đó sẽ được thêm vào danh sách phòng đang theo dõi, những phòng trong danh sách này sẽ cung cấp tính năng đến ở.



Khi nhấn vào nút đến ở ứng dụng sẽ đưa ra câu hỏi bảo mật nếu người dùng trả lời đúng câu hỏi bảo mật việc chuyển phòng mới được thực hiện

* 1. *Giao diện tạo phòng:*



* 1. *Đề xuất phòng*

