

```
.MathJax svg { shape-rendering: crispEdges; } @media print { body > div { page-break-inside: avoid; page-break-after: always; } g[style*="filter: drop-shadow("] { filter: none !important; } } Class Diagram - Student Project 01
AI-DUC.drawio.pdf@page gePageFormat-2149-1048 { margin: 0; size: 21.49in 10.48in; } .gePageFormat-2149-1048 { page: gePageFormat-2149-1048; width: 21.49in; height: 10.48in; } .gePageFormat-2149-1048 > svg { margin: 0.27in; } mjx-container[jax="SVG"] { direction: ltr; } mjx-container[jax="SVG"] > svg { overflow: visible; min-height: 1px; min-width: 1px; } mjx-container[jax="SVG"] > svg a { fill: blue; stroke: blue; } mjx-container[jax="SVG"][display="true"] { display: block; text-align: center; margin: 1em 0; } mjx-container[jax="SVG"][display="true"][width="full"] { display: flex; } mjx-container[jax="SVG"][justify="left"] { text-align: left; } mjx-container[jax="SVG"][justify="right"] { text-align: right; } g[data-mml-node="merror"] > g { fill: red; stroke: red; } g[data-mml-node="merror"] > rect[data-background] { fill: yellow; stroke: none; } g[data-mml-node="mtable"] > line[data-line], svg[data-table] > g > line[data-line] { stroke-width: 70px; fill: none; } g[data-mml-node="mtable"] > rect[data-frame], svg[data-table] > g > rect[data-frame] { stroke-width: 70px; fill: none; } g[data-mml-node="mtable"] > .mjx-dashed, svg[data-table] > g > .mjx-dashed { stroke-dasharray: 140; } g[data-mml-node="mtable"] > .mjx-dotted, svg[data-table] > g > .mjx-dotted { stroke-linecap: round; stroke-dasharray: 0,140; } g[data-mml-node="mtable"] > g > svg { overflow: visible; } [jax="SVG"] mjx-tool { display: inline-block; position: relative; width: 0; height: 0; } [jax="SVG"] mjx-tool > mjx-tip { position: absolute; top: 0; left: 0; } mjx-tool > mjx-tip { display: inline-block; padding: .2em; border: 1px solid #888; font-size: 70%; background-color: #F8F8F8; color: black; box-shadow: 2px 2px 5px #AAAAAA; } g[data-mml-node="maction"][data-toggle] { cursor: pointer; } mjx-status { display: block; position: fixed; left: 1em; bottom: 1em; min-width: 25%; padding: .2em .4em; border: 1px solid #888; font-size: 90%; background-color: #F8F8F8; color: black; } foreignObject[data-mjx-xml] { font-family: initial; line-height: normal; overflow: visible; } mjx-
```

container[jax="SVG"] path[data-c], mjax-container[jax="SVG"] use[data-c] { stroke-width: 3; }

Luồng chính, khi chạy app, việc đầu tiên được thực hiện là in/ khởi tạo logScreen & object Account. Khi có account tồn tại, khởi chạy Menu để đọc choice của user. + menuStart(boolean) : boolean

Khởi tạo account bằng input (string username, string password) Object account khởi tạo thành công <=> User được khởi tạo.

Khởi tạo account bằng input (string username, string password) Object account khởi tạo thành công <=> User được khởi tạo.

khởi tạo account bằng input (string username, string password) Object account khởi tạo thành công <=> User được khởi tạo.

khi Object User được khởi tạo và hướng các tt của người dùng hiện hành, các list cũng được khởi tạo

User thì đ tạo Report được, nên nó không có hàm newRep()
hàm đây do giải thuật kiểm duyệt của mình làm, khi duyệt thì nhả ra thì nhả vào list của user + nhả vào database.
user có thể chọn nút view để hiện ngay, hoặc tắt app mới lại vẫn xem được (2 cái hàm get sẽ chạy khi nút bấm thực hiện, đẩy đồng object từ database lên các list).

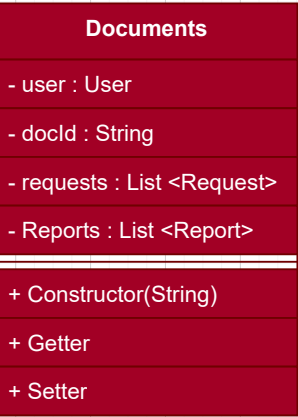
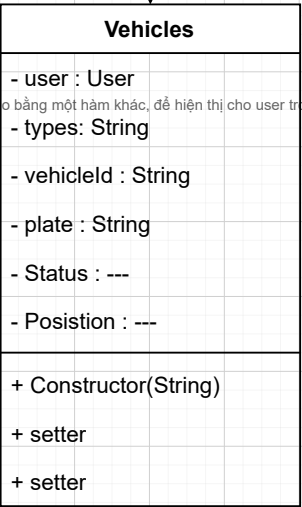
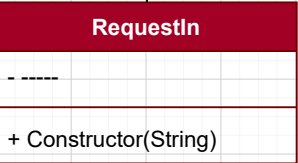
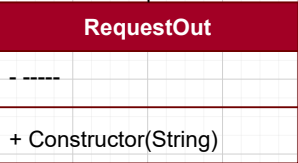
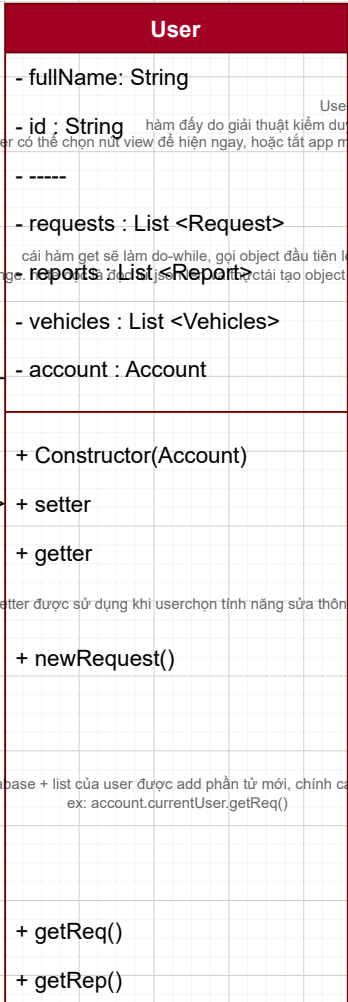
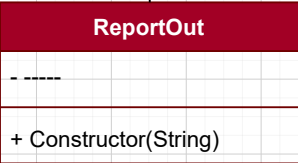
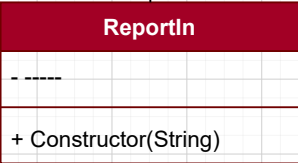
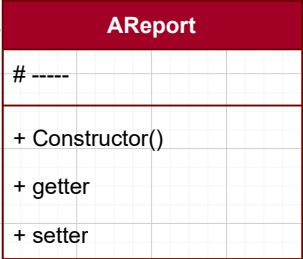
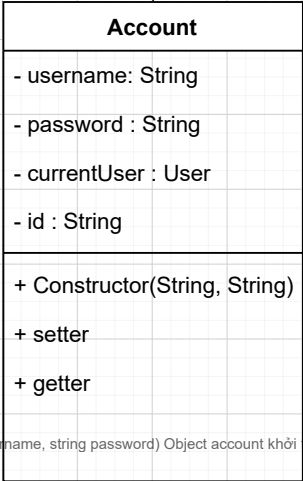
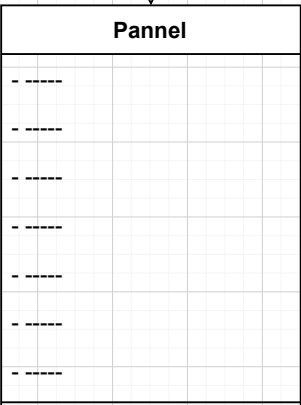
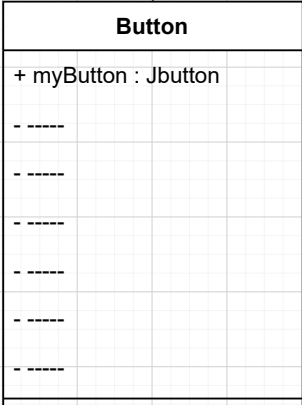
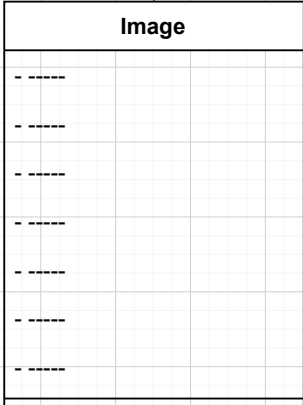
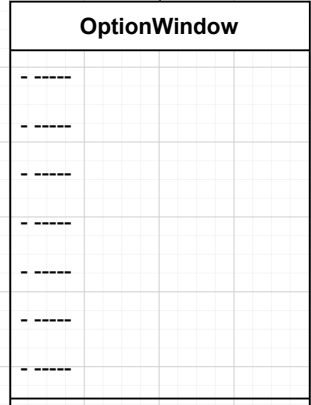
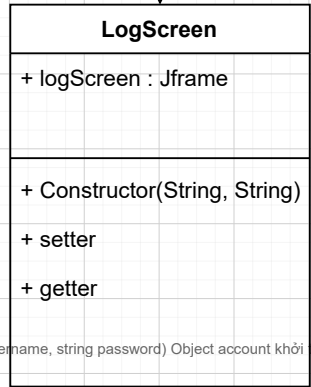
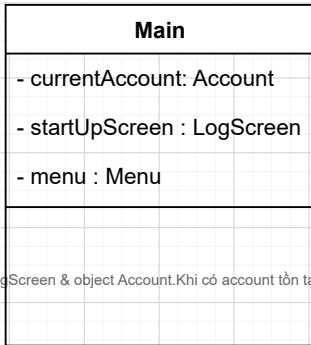
cái hàm get sẽ làm do-while, gọi object đầu tiên lên, là Null thì xem như user không có tài liệu để up lên UI. có thì check pointer tiếp theo đẩy lên tiếp.
Hoặc sai For_range, để tránh việc tái tạo object bằng dữ liệu đấy, chứ k phải đọc từ chỗ khác trong memory như thông thường (đều được tồn tại sẵn ở container khác)

User's getter setter được sử dụng khi user chọn tính năng sửa thông tin trên menu.

nhập vào database + list của user được add phần tử mới, chính cái ptu mới đó
ex: account.currentUser.getReq()

h sau khởi chạy, obj có thể được khởi tạo bằng một hàm khác, để hiển thị cho user trong trường hợp user chọn view req trên menu.

Các constructor của các object list, tồn tại bên trong object user, có quyền truy cập ID của user. Sử dụng để truy xuất dữ liệu từ database, cập nhật vào bản thân và loi ra hiển thị trên giao diện khi cần.



Quảng cho chúa ăn, thừa