Giao Việc giai đoạn 3

1. Lời nói đầu

+ Tính đến hiện tại 01/11/2025, dự án đã hoàn thiện giai đoạn nền tảng. Hướng phát triển đã có, các cơ chế như DAO interface hoặc Connection pool đã hoàn thành. Source của dự án đã có đủ 14 model cho 14 bảng trong CSDL và 14 DAO cho 14 model. Giai đoạn phát triển theo là module hay nói cách khác là xây dựng các tính năng thực tế đã có thể được triển khai. Dưới đây là chi tiết về các module mà nhóm sẽ triển khai:

+ Trước hết phải nói về nguyên tắc mà ae dùng trong dự án này:

- Làm việc song song: Các URL được quy định dưới đây là cần thiết để các module có thể phát triển độc lập. Ví dụ người phụ trách hiển thị sản phẩm ff thì sẽ chỉ cần lo hiển thị sản phẩm, chỉ cần làm một nút thanh toán giả và quy ước sẽ gọi đến "/payment/execute" chẳng hạn. Khi này 2 luồng việc có thể làm song song, người làm hiển thị không cần biết cơ chế thanh toán như nào mà chỉ cần biết pass vào cái gì đến URL nào, người làm thanh toán chỉ cần biết URL của mình được gọi đến với các tham số đã định.
- Chuyên môn hóa: đúng hơn là tách biệt logic, cùng ví dụ bên trên thì người phụ trách hiển thị sẽ chỉ cần quan tâm hiển thị cho đẹp, người làm thanh toán sẽ chỉ cần quan tâm xử lý sao cho nhanh, gọn, an toàn.
- Sản phẩm tối thiểu: Các tính năng cốt lõi sẽ được ưu tiên cao nhất để các web có thể chạy được. Sau khi test ngon lành ae có thể nâng cấp sau. Các module trình bày dưới đây tuân thủ nguyên tắc này, t chỉ viết ra những yêu cầu tối thiểu để web có thể chạy được, sau này ae muốn các cơ chế cao hơn như lọc dữ liệu hoặc phân tích dữ liệu, gacha v.v thì sẽ được triển khai sau.

2. Tổng quan các module

- Module 0: Trang chủ & Layout chung
    - Xây dựng bộ khung của web, tạo home.jsp để có cái nhìn tổng quan và cung cấp giao diện đến các module khác, header.jsp, footer.jsp để các trang khác import, đảm bảo web có bộ giao diện đồng nhất

- Module 1: Lõi người dùng & xác thực
    - Cung cấp phương thức để người dùng đăng ký, đăng nhập, đăng xuất và có cơ chế phiên đăng nhập, lọc cho các URL như thanh toán hoặc trang cá nhân khách nếu chưa đăng nhập

- Module 2: Shop acc game
    - Hiển thị danh sách các tài khoản theo danh mục, bao gồm trang tổng quan và trang chi tiết

- Module 3: Shop game Steam
    - Hiển thị danh sách các game Steam và cho người dùng xem chi tiết

- Module 4: Luồng thanh toán và trang cá nhân cho khách
    - Xử lý 2 luồng thanh toán cho tài khoản và slot game steam. Tạo một khu vực cho khách xem, sửa, xóa thông tin, xem lịch sử đơn hàng khách đặt, xem không tin tài khoản khách đã mua (cả cho tài khoản game và slot steam)

- Module 5: Trang admin
    - Xây dựng toàn bộ chức năng CRUD sản phẩm, (thêm, sửa, xóa, xem) tất cả tài khoản game, tài khoản steam, game steam, đơn hàng trên hệ thống. Và có thể là cơ chế ban tài khoản khách nếu cần (phát triển sau)
    - Xây dựng cơ chế AuthorizationFilter kiểm soát tất cả URL có dạng “/admin/*”

- Module 6: Nâng cao
    - Đúng ra cái này không hẳn là một module, nó sẽ là các chức năng hoặc cách cài đặt nâng cao cho các module trước. Nhưng vì không thể ném vào trong module cũng không thể ra ngoài, thôi thì ném vào module 6 như là một phần để ae có thể tham khảo.
    - Sau khi ae đã xong phần việc của mình thì có thể nghĩ đến nâng cấp, các module trình bày chi tiết sau đây ví dụ như module 2 sẽ chỉ yêu cầu hiển thị, một hệ thống thực tế sẽ cần có các cơ chế UX/UI như lọc theo giá, lọc theo tiêu chí, thuật toán để trình bày sản phẩm v.v

- Cái này t sẽ để đây để "nhắc nhớ" giai đoạn này sẽ tập trung triển khai cơ bản trước. Hiển nhiên là nếu trong quá trình ae làm mà nâng cấp được luôn thì tốt quá, nhưng cần cẩn trọng, một hệ thống phức tạp mà không có một nền móng tốt và ổn định sẽ rất dễ "sập".

3. Quy định URL
   - Danh sách URL sau là quy định để các luồng làm việc có thể thực hiện song song
   - Các URL pattern là quy ước cứng, bắt buộc phải theo mẫu đó cho chức năng quy định. Ae khi triển khai nếu có giải pháp khác / tên khác hãy báo sớm cho t để có thể xử lý.
   - Đối với chức năng xử lý đăng nhập và xử lý đăng kí của module 1, matKhau phải là mật khẩu đã băm 1 chiều. Nghĩa là khi người dùng điền mật khẩu để đăng nhập hoặc đăng kí, mật khẩu này sẽ được băm trực tiếp và chỉ gửi đi để xử lý với CSDL mật khẩu đã băm. Người dùng hoặc admin không thể xem mật khẩu khi đã băm 1 chiều, khi người dùng đăng nhập, mật khẩu họ nhập sẽ được băm và so khớp chuỗi đã băm với mật khẩu (đã băm) lưu trong CSDL. Điều này đặt ra 1 vấn đề là khi người dùng quên mật khẩu, cần có cơ chế lấy lại qua mail hoặc sdt (cái này để trong module 6, cài đặt sau)
   - Đối với mật khẩu của các tài khoản game và tài khoản steam, khi vận chuyển và lưu trong CSDL phải là dạng băm 2 chiều. Nghĩa là khi admin nhập mật khẩu cho tài khoản, backend thực hiện cơ chế băm 2 chiều và chỉ gửi đi để xử lý và lưu trong CSDL mật khẩu đã băm. Khi người dùng hoặc admin cần xem lại mật khẩu, backend truy vấn mật khẩu từ CSDL (dạng đã băm) thực hiện giải mã và đưa ra mật khẩu ban đầu.
   - Hiển nhiên danh sách trên được đưa ra có thể không phản ánh hết các chức năng hoặc phản ánh sai, thừa. Điều này cần ae phải phối hợp chặt chẽ với me (Hưng) và với bất kỳ thay đổi, góp ý ngoài kế hoạch, rất mong ae có thể thông báo sớm cho t, t sẽ luôn sẵn sàng để làm việc với ae.

Module	Chức năng	Method	URL (pattern)	Tham số (Query Params)
0. Home	Trang chủ	GET	/home	
1. User	Hiển thị form Đăng nhập	GET	/login	
	Xử lý Đăng nhập	POST	/login	tenDangNhap, matKhau
	Hiển thị form Đăng ký	GET	/register	
	Xử lý Đăng ký	POST	/register	tenDangNhap, matKhau, email, v.v.
	Xử lý Đăng xuất	GET	/logout	
2. Shop Game	Danh sách (Lọc)	GET	/shop/game	category (vd: lienquan, freefire)
	Chi tiết	GET	/shop/game/detail	id (MaTaiKhoan), category
3. Shop Steam	Danh sách	GET	/shop/steam	
	Chi tiết	GET	/shop/steam/detail	id (MaGameSteam)
4. Payment	Xử lý thanh toán	GET	/payment/execute	type (game|steam), id (MaTaiKhoan | MaGameSteam)
	Trang thành công	GET	/payment/success	orderId, type
5. Profile	Trang cá nhân	GET	/profile	
	Sửa thông tin (Xử lý)	POST	/profile/edit	(Các trường NguoiDung)
	Lịch sử đơn hàng	GET	/profile/orders	
	Xem TK đã mua	GET	/profile/view-account	orderId, type (game|steam)
6. Admin	(Prefix)	ALL	/admin/*	
	Dashboard	GET	/admin/dashboard	
	Quản lý Users (List)	GET	/admin/users	
	Quản lý Users (Action)	POST	/admin/users	action (create, update, delete)
	Quản lý Đơn hàng	GET	/admin/orders	
	Cập nhật Đơn hàng	POST	/admin/orders/update	orderId, type, trangThai

4. Chi tiết yêu cầu cho các module
   - Các yêu cầu cho module là tối thiểu để cho tính năng có thể chạy được. Người phụ trách phải đảm bảo đáp ứng các yêu cầu được đề ra, bao gồm cả các test để đảm bảo chất lượng. Phần chi tiết cài đặt mang tính tham khảo, khuyến khích ae đưa ra giải pháp của riêng mình hoặc phát triển dựa trên đề xuất của t.

- Module 0: Trang chủ & Layout chung
  - Tạo thư mục mới: src/main/webapp/common/. Đặt 2 file header.jsp và footer.jsp vào trong đó.
  - Tạo thư mục mới: src/main/webapp/assets/ (và các thư mục con như css/, js/, images/).
  - Tạo file assets/css/style.css (định nghĩa CSS chung) và assets/js/script.js (JS chung).
  - Tạo home.jsp (đặt tại src/main/webapp/home.jsp).
  - Tạo HomeServlet (map với /home) để gọi DAO và forward ra home.jsp.
  - Tất cả các JSP khác (ví dụ: login.jsp, profile.jsp) phải dùng <jsp:include page="common/header.jsp" /> và <jsp:include page="common/footer.jsp" />.

- Module 1: Lõi người dùng & xác thực
  - Tạo login.jsp và register.jsp (đặt tại src/main/webapp/).
  - Tạo LoginServlet (map với /login) để xử lý logic đăng nhập, tạo session session.setAttribute("user", nguoiDung).
  - Tạo LogoutServlet (map với /logout) để hủy session.
  - Tạo RegisterServlet (map với /register) để xử lý đăng ký.
  - Tạo AuthenticationFilter (Filter):
    - Map filter này với các URL: /profile/* và /payment/*.
    - Kiểm tra nếu session.getAttribute("user") == null, thì chuyển hướng về /login.

- Module 2: Shop acc game – kiến trúc dùng chung 1 site cho 3 loại tk
    - Tạo taikhoangame.jsp (danh sách) và taikhoangame_chitiet.jsp (chi tiết) (đặt tại src/main/webapp/).
    - Tạo TaiKhoanGameServlet (map với /shop/game và /shop/game/detail).
    - Luồng Danh sách (GET /shop/game): Đọc category, gọi DAO con tương ứng (ví dụ: TaiKhoanLienQuanDAO.getByTrangThai("DANG_BAN")), forward ra taikhoangame.jsp.
    - Luồng Chi tiết (GET /shop/game/detail): Đọc id và category. Gọi DAO con tương ứng (ví dụ: TaiKhoanLienQuanDAO.getById(id)) và AnhTaiKhoanDAO.getByMaTaiKhoan(id). Gửi cả 2 (Tài khoản và List Ảnh) ra taikhoangame_chitiet.jsp.
    - taikhoangame_chitiet.jsp phải có nút "Mua Ngay" trỏ đến URL Module 4: /payment/execute?type=game&id=${taiKhoan.maTaiKhoan}.

- Module 3: Shop game steam
    - Tạo steam_games.jsp (danh sách) và steam_game_detail.jsp (chi tiết) (đặt tại src/main/webapp/).
    - Tạo SteamGameServlet (map với /shop/steam và /shop/steam/detail).
    - Luồng Danh sách (GET /shop/steam): Gọi GameSteamDAO.fastGetAll(). Forward ra steam_games.jsp.
    - Luồng Chi tiết (GET /shop/steam/detail): Đọc id (MaGameSteam). Gọi GameSteamDAO.getById(id) VÀ BaiVietGioiThieuDAO.getByMaGameSteam(id) . Gửi cả 2 đối tượng ra steam_game_detail.jsp.
    - steam_game_detail.jsp phải có nút mua slot trỏ đến URL Module 4: /payment/execute?type=steam&id=${gameSteam.maGameSteam}.

- Module 4: Thanh toán & trang cá nhân cho khách
    - Tạo thư mục mới: src/main/webapp/profile/.
    - Tạo các file JSP: profile/view.jsp, profile/orders.jsp, profile/view_account_details.jsp và payment_success.jsp (đặt tại src/main/webapp/).
    - Tạo PaymentServlet (map với /payment/execute và /payment/success):
        - Luồng acc game : Có thể viết một hàm mới trong TaiKhoanDAO kiểm tra xem sản phẩm này có trong đơn hàng nào có trạng thái là “CHO_THANH_TOAN” và “DA_HOAN_THANH” không, nếu không thì mới cho phép thanh toán. Thực hiện chèn một bản ghi đơn hàng với trạng thái “CHO_THANH_TOAN” nếu chưa có hoặc là kéo bản ghi đang thanh toán dở lên (với thời gian còn lại) và đợi người dùng thanh toán. phải kiểm tra và so khớp mã khách hàng trong session với trong đơn hàng đang thanh toán dở.
        - Luồng slot steam :
            - Nếu không có đơn hàng nào còn hiệu lực từ trước: Thực hiện logic cân bằng tải, có thể viết một hàm mới trong TaiKhoanSteamDAO hoặc GameTaiKhoanSteamDAO hoặc GameSteamDAO để thực hiện JOIN các bản ghi trong GAME_TAIKHOAN_STEAM với TAIKHOAN_STEAM để lấy ra bảng các tài khoản cho một game steam, thực hiện lọc theo (TongSoSlot – SoSlotDaBan) đảm bảo khách sẽ luôn mua ở tài khoản ổn định nhất (còn nhiều slot nhất) và có thể trả về cả dãy hoặc 1 tài khoản đầu danh sách. Sau khi chọn được tài khoản cho khách, thực hiện tăng SoSlotDaBan cho tài khoản đó tạm thời lên một, đồng thời chèn một bản ghi vào DONHANG_SLOT_STEAM với trạng thái “CHO_THANH_TOAN” và đợi người dùng thanh toán.
            - Nếu có đơn hàng còn hiệu lực thì đơn giản là kéo lên và đợi khách thanh toán thôi (với thời gian còn lại), và hiển nhiên, phải kiểm tra và so khớp mã khách hàng trong session với trong đơn hàng.
            - Nếu đơn hàng bị hủy, thực hiện khôi phục số slot cho tài khoản tương ứng.
            - Vì triển khai trên local nên khả năng sẽ không có cơ chế thanh toán tự động, vậy nên ae sẽ làm là hiện QR đến video này Rick Astley - Never Gonna Give You Up (Official Video) (4K Remaster) và hiện số tiền hoặc một số thông tin làm màu khác. Admin sẽ xem đơn và trực tiếp xác nhận hoặc từ chối.
        - Chuyển hướng (redirect) đến /payment/success nếu thành công

        - Thực hiện cơ chế kiểm soát găm hàng thông qua thuộc tính ThoiGianTao của DONHANG và DONHANG_SLOT_STEAM, sử dụng thuộc tính này để set timeout cho một đơn hàng hoặc có cơ chế quét sau một thời gian nhất định.
    - Tạo UserProfileServlet (map với /profile và /profile/edit): Hiển thị và cập nhật thông tin NguoiDung.
    - Tạo OrderHistoryServlet (map với /profile/orders): Gọi DonHangDAO.getByMaNguoiDung() VÀ DonHangSlotSteamDAO.getByMaNguoiDung() và gửi 2 danh sách ra profile/orders.jsp.
    - Tạo ViewPurchasedAccountServlet (map với /profile/view-account): Xử lý logic hiển thị tài khoản/mật khẩu cho các đơn hàng đã DA_HOAN_THANH.

- Module 5: Trang quản trị của admin
    - Tạo thư mục mới: src/main/webapp/admin/. Tất cả các JSP của Admin (ví dụ: admin/dashboard.jsp, admin/manage_users.jsp, admin/manage_products.jsp) phải đặt trong này.
    - Tạo AuthorizationFilter (Filter):
        - Map filter này với URL /admin/*.
        - Kiểm tra NguoiDung trong session, nếu user == null HOẶC !user.getVaiTro().equals("ADMIN"), chuyển hướng về /home.
    - Tạo các Servlet CRUD (ví dụ: AdminUserServlet, AdminProductServlet, AdminOrderServlet) và map với prefix /admin/* (ví dụ: /admin/users, /admin/products/edit).
    - Servlet quản lý đơn hàng:
        - Hủy đơn game: cập nhật lại trạng thái cho đơn hàng để admin có thể chặn hoặc xác nhận thanh toán cho khách. Có thể gọi sang các cơ chế hủy thanh toán của module 4
        - Hủy đơn Steam: Cập nhật trạng thái đơn để ban hoặc xác nhận. Có thể gọi sang các cơ chế hủy thanh toán của module 4 để tự động hồi trả slot nếu bị admin hủy đơn.

5. Những vấn đề về phụ thuộc

+ Dựa trên danh sách các module trên, có thể nhận thấy rằng:

- Module 1, 2, 3, 4, 5 phụ thuộc module 0: tất cả module đều cần header.jsp và footer.jsp
- Module 4, 5 phụ thuộc module 1: Module 4, 5 cần đối tượng người dùng
- Module 4 phụ thuộc vào module 2 & 3: thanh toán thì cần nút đến thanh toán

- Giải quyết cho vấn đề thứ nhất, t sẽ cố gắng làm một homepage và tạo các thư mục "src/main/webapp/common/" và "src/main/webapp/assets/css/", tạo các file header.jsp, footer.jsp, style.css nhanh nhất có thể.

- Đối với vấn đề thứ hai, việc module thanh toán và trang cá nhân cho khách, admin cần có tài khoản trước. Ở đây t sẽ đề xuất 1 giải pháp là làm filter fake, filter này sẽ tạo một admin giả và nhét vào session để bypass AuthorizationFilter.

- Đối với vấn đề nút bấm thì lại đơn giản, chỉ cần kiểm thử servlet của mình bằng cách gõ thẳng đến URL định trước kèm các tham số quy ước. Ví dụ:
  - http://localhost:8080/microShop/payment/execute?type=game&id=1
  - http://localhost:8080/microShop/payment/execute?type=steam&id=2
  - Trong đó "payment/execute" là URL đã định trước, type và id là tham số quy ước

6. Chia việc và deadline
   - Tuấn Anh : Module 1
   - Hải : Module 2
   - Thành : Module 3
   - Khoa : Module 4
   - Hưng : Module 0 & Module 5

- Lưu ý vì thầy không cho phép ae chia ra frontend và backend nên chia theo module như này có nghĩa là ae sẽ phải làm từ a-z. Ví dụ đối với module của Tuấn Anh thì sẽ phải có cả logic đăng nhập, đăng ký, đăng xuất và cả phần frontend cho các logic đấy luôn bao gồm trang đăng nhập, nút đăng nhập ở trang chủ v.v. Miễn làm sao khi ae báo cáo cho me (Hưng) thì t chỉ cần bấm vào phần ae làm, có nút bấm, giao diện như web thật và với tất cả chức năng yêu cầu là đc.

- Vì phần này khá đặc biệt, chỉ có yêu cầu về kết quả chứ không có yêu cầu các đầu việc phải làm như các phần khác. Vậy nên dealine sẽ là “best-effort” dù sao đây cũng là giai đoạn gần cuối rồi và lại là giai đoạn quan trọng nhất, không chỉ định hình bài tập lớn của một môn học, mà còn là bước chân đầu tiên cho ae đến với con đường làm dev. Vậy nên ae sẽ phải cố gắng thật nhiều, nhiều nhất trong khả năng để làm phần chức năng của mình.

- Đối với phần module này ae sẽ không làm theo deadline mà ae sẽ báo cáo me thường xuyên về những phần ae làm được, cũng như liên hệ me khi ae cần tư vấn, hỗ trợ v.v. Cứ theo khuôn 4-5 ngày ae sẽ có một cuộc họp tiến độ để t xem xét tổng thể tình hình dự án và ra quyết định. Vậy thôi, chúc ae làm việc hiệu quả!