1. Kiến trúc phần mềm

Database

Data Direct Access Layer (Jar lib/Hibernate/Transactional)

Business Logic/Business Rules/Security (Jar lib)

Internal Services Bus (RESTful/JSON/WSO2)

Security verification (Authentication)

Presentation Layer – Internal Usage (LAN/WAN/Angular)

Data/Business Validation (Jar lib)

External Security check

External Services Bus (File/JSON/XM/L/ WSO2)

Presentation Layer – External Usage (Mobile/Internet/Angular)

Schedule Job

3rd Parties

Other MC’s Systems

Security check (Authorization/Jar lib)

1. Coding guideline:
2. Lớp giao diện
3. Lớp service

* Dựa trên class/method ở lớp nghiệp vụ đã định nghĩa sẵn input/output, lớp service định nghĩa input/output theo format JSON cho input/output ở lớp nghiệp vụ

1. Lớp bảo mật (xác thực, mã hóa)

* Xác thực với AD
* Giãi mã/mã hóa message nếu có yêu cầu

1. Lớp core logic nghiệp vụ
   1. Lớp bảo mật (cấp phép)

* Nếu service login thì kiểm tra phân quyền của user , trả ra thông tin quyền hạn được truy cập của user, đẩy thông tin vào bảng session
* Kiểm tra phân quyền xem người dùng có được quyền truy cập chức năng hiện tại không
  1. Lớp kiểm tra tính hợp lệ
* Kiểm tra tính hợp lệ của dữ liệu đầu vào
  1. Lớp xử lý logic nghiệp vụ
* Xử lý nghiệp vụ theo yêu cầu của chức năng

1. Lớp truy cập cơ sở dữ liệu

* Truy cập CSDL qua Hibernate

1. a
2. Coding convention:

* Tên biến đặt theo convention của Java
* Tên các biến private của class bắt đầu bằng chữ thường, sau đó upper case chữ cái đầu của các từ tiếp theo. Ví dụ: private String customerCode;
* Sử dụng công cụ phát triển (Eclipse) để đặt tên cho các getter, setter của biến private, đảm bảo đúng chuẩn
* Tên class bắt đầu bằng chữ hoa chữ cái đầu
* Tên các class define objects chứa data bắt đầu bằng OBxxxx
* Lớp Validation: Nguyên tắc tạo class, method: validate cho mỗi màn hình/chức năng sẽ nằm trong cùng 1 class, validate cho từng trường thuộc 1 màn hình/chức năng sẽ là 1 method trong class đó, nhiều rule validate khác nhau cho cùng 1 trường sẽ là nhiều method khác nhau
* Lớp Business: Đặt tên class theo các chủ đề nghiệp vụ như customer, credit, product, card …

1. Cấu trúc các lớp thư viện thành phần
2. Thư viện truy cập cơ sở dữ liệu:

Tên thư viện: **mcdataaccess.jar**

Các packages:

|  |  |
| --- | --- |
| Tên package | Mô tả chức năng |
| com.mcredit.data.customer |  |
| com.mcredit.data.loan |  |
| com.mcredit.data.zzzz |  |
| com.mcredit.common |  |

1. Thư viện bao gồm các xử lý logic, quy tắc nghiệp vụ, phân quyền, kiểm tra hợp lệ:

Tên thư viện: **mcbusiness.jar**

Các packages:

|  |  |
| --- | --- |
| Tên package | Mô tả chức năng |
| com.mcredit.business.customer |  |
| com.mcredit.business.product |  |
| com.mcredit.security.authentication |  |
| com.mcredit.security.authorization |  |
| com.mcredit.validation.customer |  |
| com.mcredit.validation.contract |  |
|  |  |
|  |  |

1. Thư viện cho các service sử dụng

Tên thư viện: **mcservice.jar**

Các packages:

|  |  |
| --- | --- |
| Tên package | Mô tả chức năng |
| com.mcredit.service.customer |  |
| com.mcredit.service.security |  |