CHAPTER 3:

SOFTWARE ARCHITECUTRE

3.1. SYSTEM OVERVIEW

Warehouse Management System is web-based application designed to support and optimize warehouse or distribution center management. They facilitate management in their daily planning, organizing, staffing, directing, and controlling the utilization of available resources, to move and store materials into, within, and out of a warehouse, while supporting staff in the performance of material movement and storage in and around a warehouse.

3.2. SYSTEM ENTITIES

Table 3.6 System Entities

ID	Entities
E01	Director
E02	Administrator
E03	Supply Chain Department
E04	Manage Project Department
E05	Design Department
E06	QAQCQS
E07	Site Manager
E08	Stocker
E09	Email Server

3.3. HIGH-LEVEL FUNCTIONAL REQUIREMENTS

Table 3.7 High-level functional requirements

ID	FUNCTIONAL REQUIREMENT
F01	Manage User Account
F02	Manage Permission for Action and Data
F03	Manage Catalog to Classify Material
F04	Manage Material
F05	Manage Import Operation
F06	Manage Export Operation
F07	Manage Quotation
F08	Manage Purchasing
F09	Manage Material Request

3.4. QUALITY ATTRIBUTES

Table 3.3 List of Quality Attribute

No.	ID	Quality	Brief description
1	Q01P01	Performance	Response time for queries data
2	Q01P02	Performance	Transfer detail list
3	Q02S01	Security	Control system accessing
4	Q02S02	Security	Control data accessing
5	Q02S03	Security	Control action accessing
6	Q03U01	Usability	Easy to use
7	Q04M01	Modifiability	Easy to modify

3.4.1. Performance Scenario

Table 3.8. Performance Scenario 01

Q01P01	**Response time for queries data, especially when users select material in Detail list**
Source	Stocker, Site manager, Supply Chain Department
Stimulus	User need to select a lot of materials in the detail list
Environment	Stocker, SM use construction Wi-Fi which is unstable SCD is in normal condition
Artifact Stimulate	Purchase Receipt Detail, Quotation Receipt Detail, Request Receipt Detail, Import Receipt Detail, Export Receipt Detail
System response	System loads all information that business required before the user querying
Response measure	5 second for loading in construction Wi-Fi 2 second for loading in normal condition

Q01P02	**Transfer material in detail list to another detail list for other business purpose instead of entering the materials again**
Source	Stocker, Site manager, Supply Chain Department
Stimulus	User need to transfer a lot of materials in the detail list to anther detail list
Environment	Stocker, SM use construction Wi-Fi which is unstable SCD is in normal condition
Artifact Stimulate	Import Receipt Detail, Export Receipt Detail, Purchase Receipt Detail, Quotation Receipt Detail, Request Receipt Detail
System response	System copy material from detail list into the new Receipt user
Response measure	5 seconds for loading in construction Wi-Fi 2 seconds for loading in normal condition

3.4.2. Security Scenario

Table 2.9 Security Scenario 01

Q02S01	**The need to control access to the system**
Source	Undefined access
Stimulus	There is an undefined access that attempt to access the system
Environment	Normal condition
Artifact Stimulate	Module Gateway
System response	System check user token in Security tier If value returned is true, user is allowed to access the Server tier Else, system shows alert for not defined accessing
Response measure	

Table 3.10 Security Scenario 02

Q02S02	**The need to control access by data attribute**
Source	User
Stimulus	User want cannot access denied data
Environment	Normal condition
Artifact Stimulate	System data permission
System response	System check the JSON of permission and show data accompany to the result, whom permission is denied cannot view the data
Response measure	

Table 3.11 Security Scenario 03

Q02S03	**The need to control access based on system function**
Source	Undefined access
Stimulus	User want to access denied action
Environment	Normal condition
Artifact Stimulate	System action permission
System response	System check the JSON of permission and show data accompany to the result, whom permission is denied cannot do the action
Response measure	

Table 3.12 Usability Scenario

Q03U01	**The store operation management is easy and pleasant to use. It is also has clear manipulations for saving time.**
Source	Site manager, Stocker
Stimulus	SM, Stocker don’t have time for complex manipulations, so that the usability quality is defined by 5 quality components: Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design? Efficiency: Once users have learned the design, how quickly can they perform tasks? Memorability: When users return to the design after a period of not using it, how easily can they reestablish proficiency? Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors? Satisfaction: How pleasant is it to use the design?
Environment	Mobile platform and website
Artifact Stimulate	User interface and the processing flow
System response	Collect user feedback to improve the usability quality
Response measure	

3.4.3. Modifiability Scenario

Table 3.13 Modifiable Scenario

Q04M01	**The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment.**
Source	Red sun developers
Stimulus	Simplification - streamlining/simplifying functionality Restructuring - rationalizing services, modularizing/optimizing/creating reusable components Functional scalability - ability to scale both up/down in terms of users, system throughput, availability, etc. Functional flexibility - turning an existing capability to new uses, new locations, or unforeseen situations Component complexity - in general the more complex the components, the more difficult they are to change Component size - smaller components are generally easier to modify than large ones Scope of modification - architecture level modifications are more difficult; may involve a complete redesign with different components and interactions
Environment	Normal condition
Artifact Stimulate	Warehouse system source code
System response	Partition a system into distinct modules representing separate areas of functionality
Response measure	1 developer's task has less conflict with others

3.5. SYSTEM CONSTRAINT

3.5.1. Business Constraint

The team will utilize the Architecture Centric Design Method to development Requirement and Architecture Design.

The team will utilize the Kanban for Implementation Phase.

The team has 30 weeks to develop the project with committed requirement.

At first semester, everyone spends 5 hours/day – 7days/week.

At second semester, everyone spends 8 hours/day – 6 days/week, 5 hours for Saturday.

3.5.2. Technical Constraint

Framework: Spring MVC, AngularJS, Bootstrap, Hibernate

Database: PostgreSQL

Default browser: Google Chrome

IDE: Eclipse, Visual Studio Code

ADE: Postman

3.6. ARCHITECTURE OVERVIEW

3.6.1. Pattern versus Tactic

Software architecture designers inevitably work with both architecture patterns and tactics. Architecture patterns describe the high-level structure and behavior of software systems as the solution to multiple system requirements, whereas tactics are design decisions that improve individual quality attribute concerns. Tactics that are implemented in existing architectures can have significant impact on the architecture patterns in the system. Similarly, tactics that are selected during initial architecture design significantly impact the architecture of the system to be designed: which patterns to use, and how they must be changed to accommodate the tactics.

© 2010 Elsevier Inc

3.6.2. Module View Definition

Modules pertain to the way in which a system’s software is decomposed into manageable units of responsibilities, which is one of the important forms of system structure. Modules are related to one another by forms of is-part-of, depends-on, and is-a relations. A module view provides a blueprint for the source code and the data model. Document module interfaces to establish a module’s role in the system. Module views are commonly mapped to component-and connector views. In general, a module may participate in many runtime components.

Documenting Software Architectures

3.6.3. Component and Connector Views

Component-and-connector views describe structures consisting of elements that have runtime presence, such as processes, objects, clients, servers, and data stores. Additionally, C&C views include as elements the pathways of interaction, such as communication links and protocols, information flows, and access to shared storage. Components have interfaces, which are called ports.

Documenting Software Architectures

3.6.4. Allocation Views

Allocation styles map software elements to elements in the environment of the software.
A deployment view describes the mapping of runtime software elements to the hardware of the computing platform on which the software executes.
An install view describes the tree structure of files and folders in the production environment and how the software components are mapped to that structure.
A work assignment view describes the mapping of modules onto the people, groups, or teams tasked with the development of those modules.

Documenting Software Architectures

3.7. CONTEXT DIAGRAM

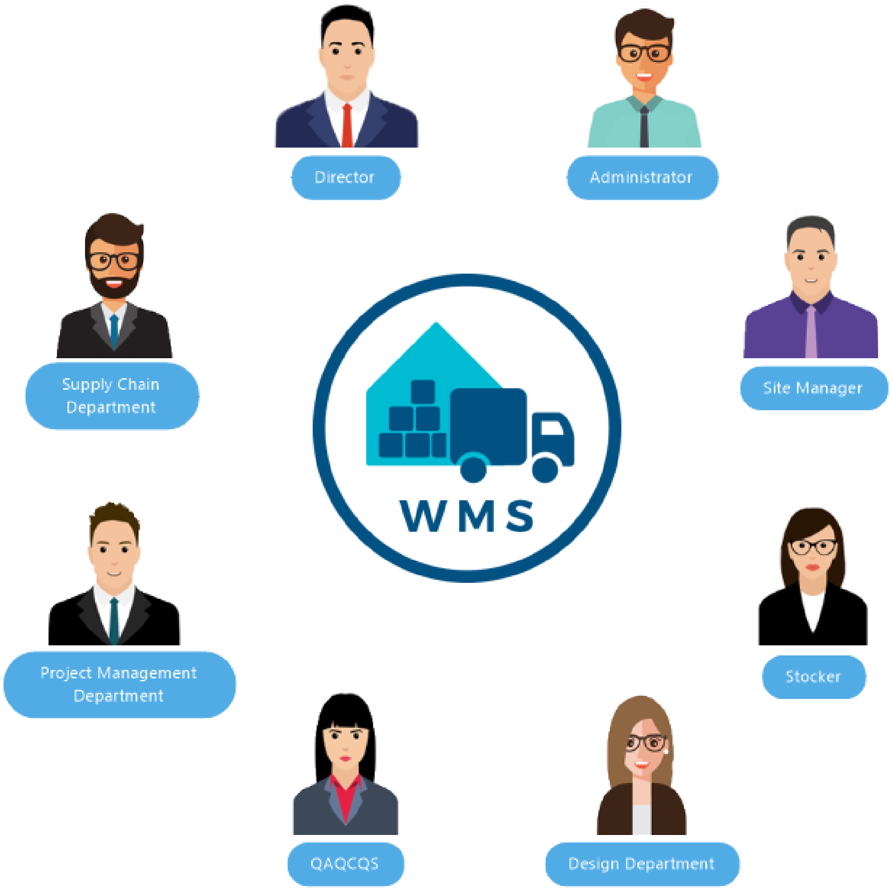

Figure 3.5 Context Diagram

3.8. ALLOCATION VIEW

In the Allocation View, RECKLESS designed the view base on Deployment Style, In the deployment style, software elements native to a C&C style are allocated to the hardware of the computing platform on which the software executes.

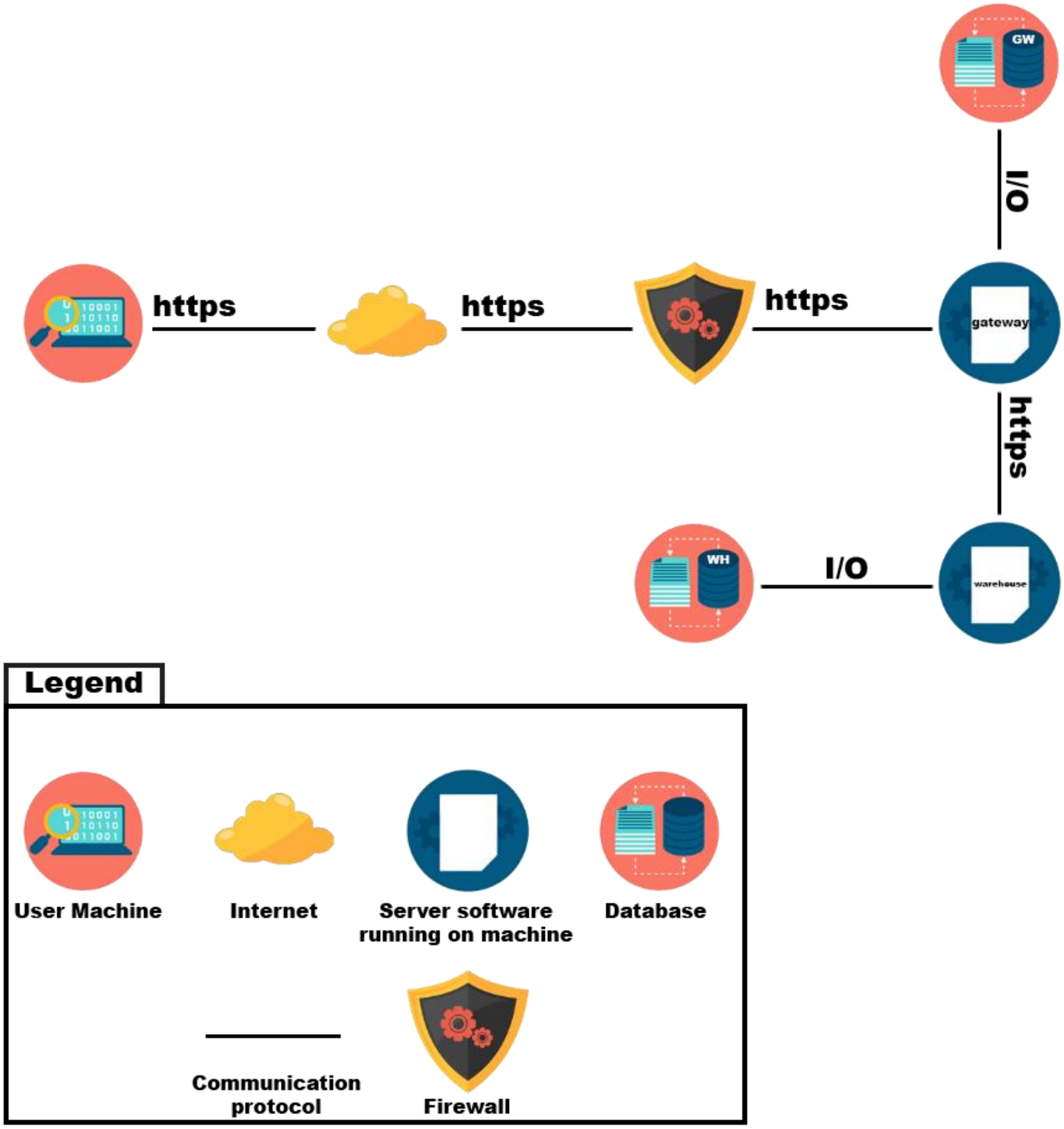

Figure 3.6 Allocation View

Table 3.10 Element of Allocation View

Element	Type	Description
https	Protocol	Using https protocol to encrypt all the communications between the client browser and website
Firewall	Software/ Hardware	This element for father deployment. Firewall to recognize damage access to the website and user behavior
Gateway	Software	This software is to recognize user access for allowing them to access the system
Warehouse	Software	This software is main application for managing warehouse
GW database	storage	Store user information to authenticate
WH database	storage	Store warehouse information

3.9. C&C VIEW

In the C&C View, RECKLESS designed the view base on Multi-tier Style, which is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms.

Multi-tier Style is able to distribute into many layers over multiple physical tiers. This can make a very good impact on improving performance, scalability and fault tolerance. The maintenance is also easier because of the low coupling between layers and adding more functionality to your application is made easier.

RECKLESS team uses tactic Control Resource Demand and Manage Resource to achieve the Performance quality.

In the Control Resource Demand, we use Prioritize events tactic. This means the performance for querying detail lists are ordered accompany the priority: Import receipt, Export receipt, Request receipt, Quotation receipt, Purchase receipt

3.9.1 Project Overview

- Level 1

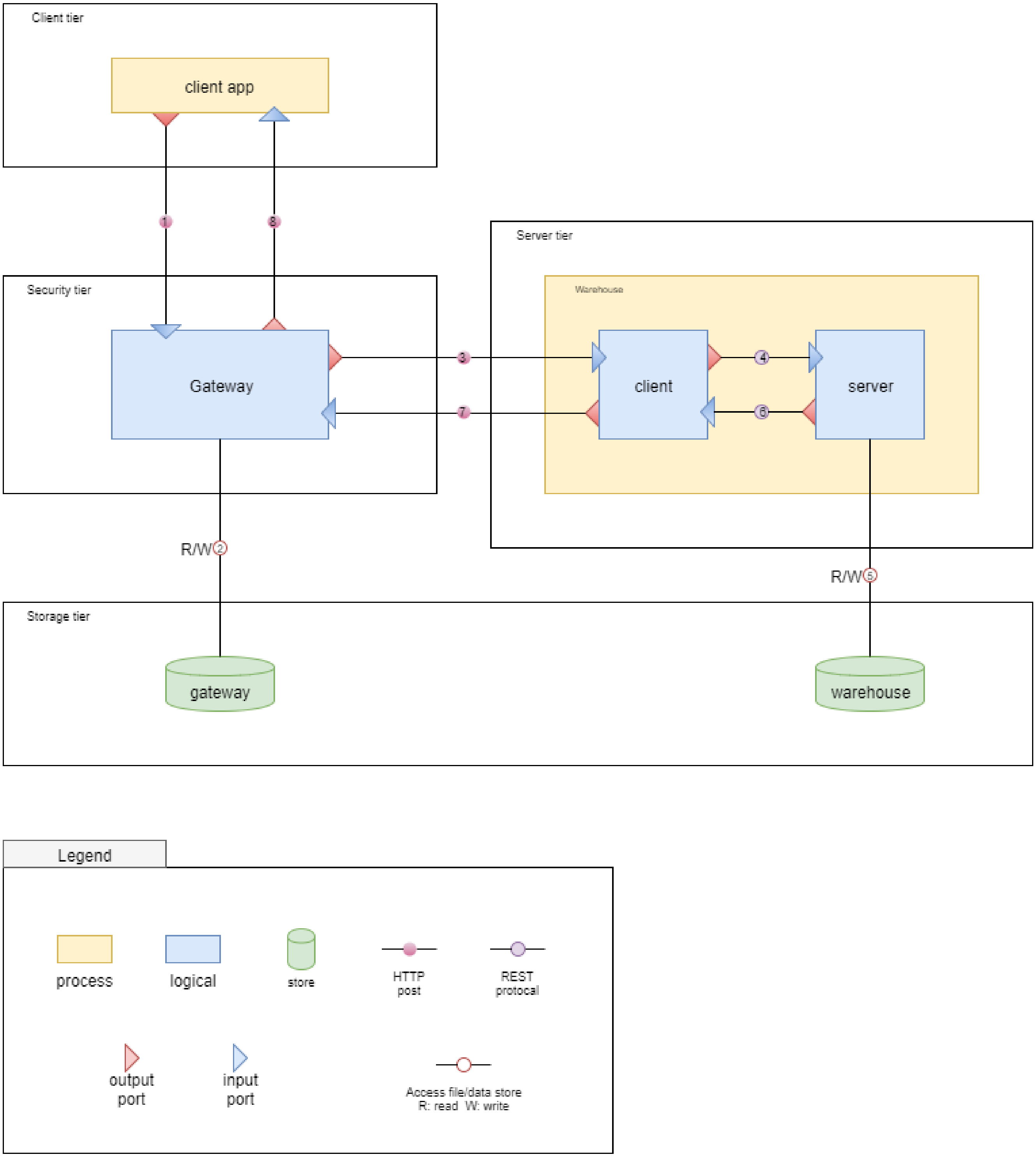

Figure 3.7 C&C View - Project Overview Level 1

- Level 2

This Level 2 is the decomposition of Warehouse module

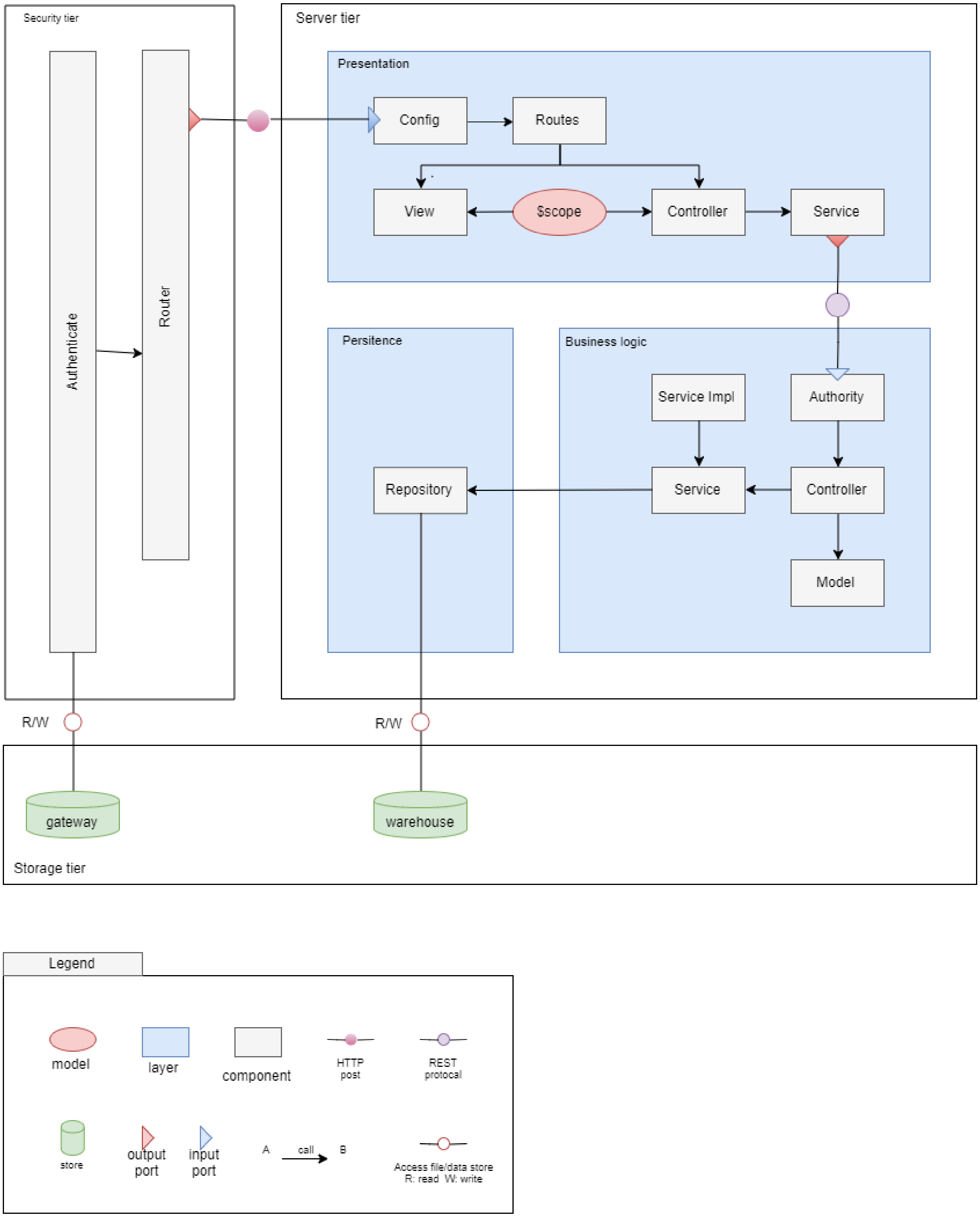

Figure 3.8 C&C View - Project Overview Level 2

Table 3.11 Element of C&C View

Element	Type	Description
Authentication	Layer	Authenticate user to access the system, return token that mark to remember allowed user
Router	Layer	Direct user to the module that allow to access
Route	Component	Direct user to the view that they access
View	Component	An output representation of information
$scope	Model	Binding part between view and controller
Controller presentation layer	Component	Control the data of AngularJS application
Service presentation layer	Component	Provide interface to communicate with business layer
Authority	Component	Component which provide permission on data and action to user
Controller business layer	Component	The controller responds to the user input and performs interactions on the data model objects. The controller receives the input, optionally validates it and then passes the input to the model.
Service business layer	Component	Interface of Service Implement
Service Impl	Component	Execute business logic
Repository	Component	Store object to persisted or retrieved from some storage
Model	Component	Encapsulates the application data

3.9.2. User Login to System

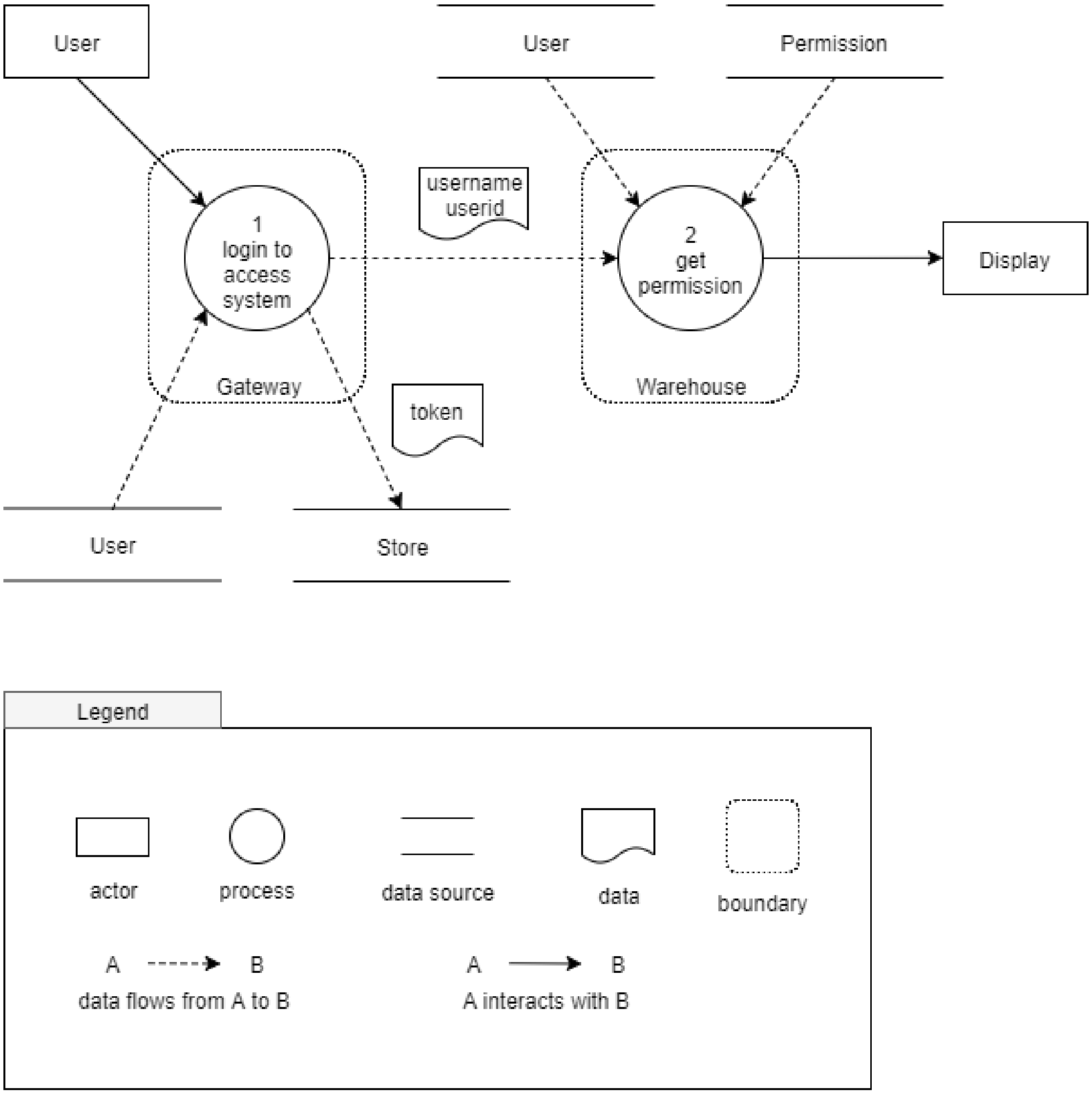

Figure 3.9 C&C View - User login to system

3.9.3. Site Manager send Request Receipt to SCD

- Level 1

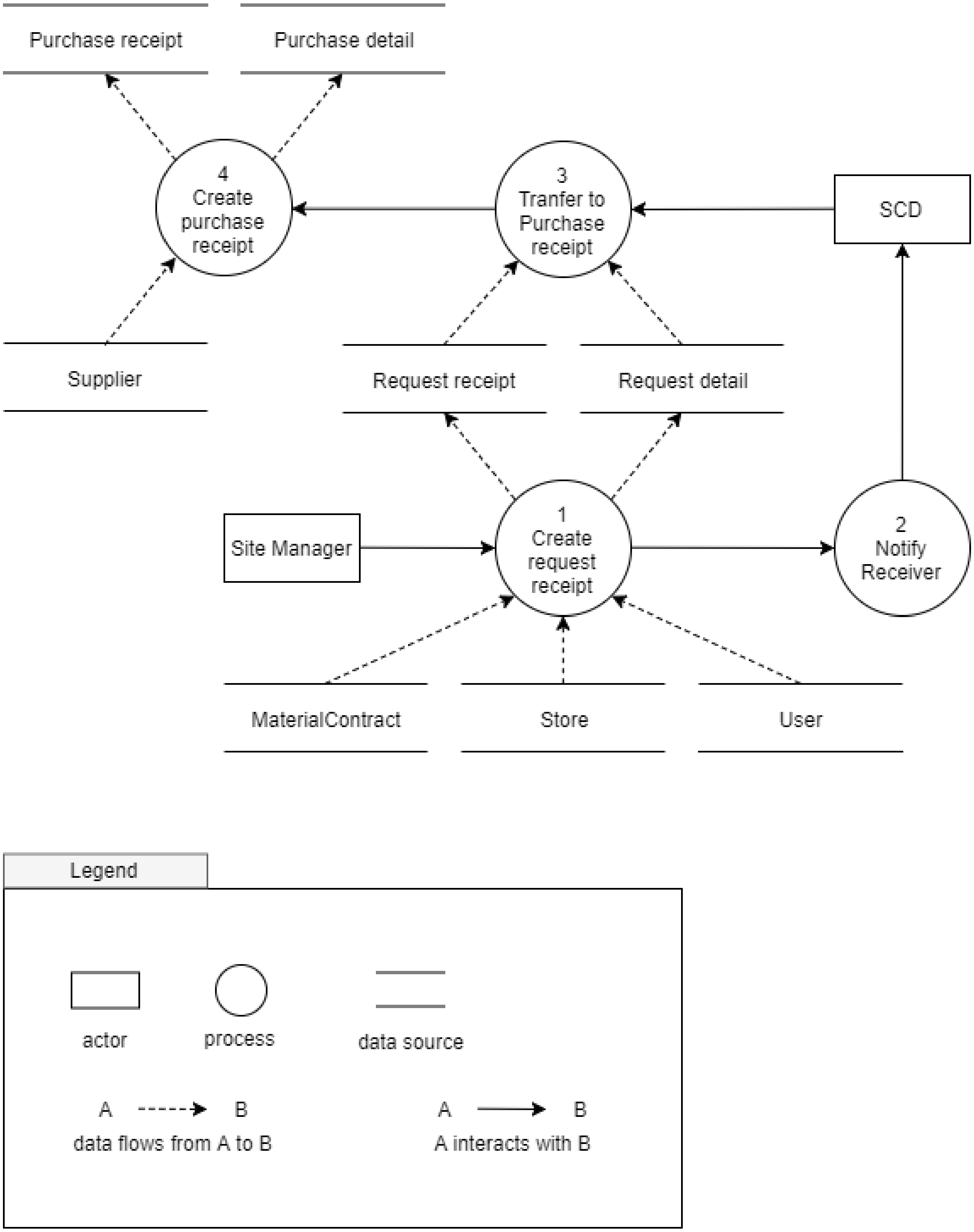

Figure 3.10 Site Manager send Request Receipt to SCD

- Level 2

This Level 2 is the decomposition of process 1 Create Request receipt

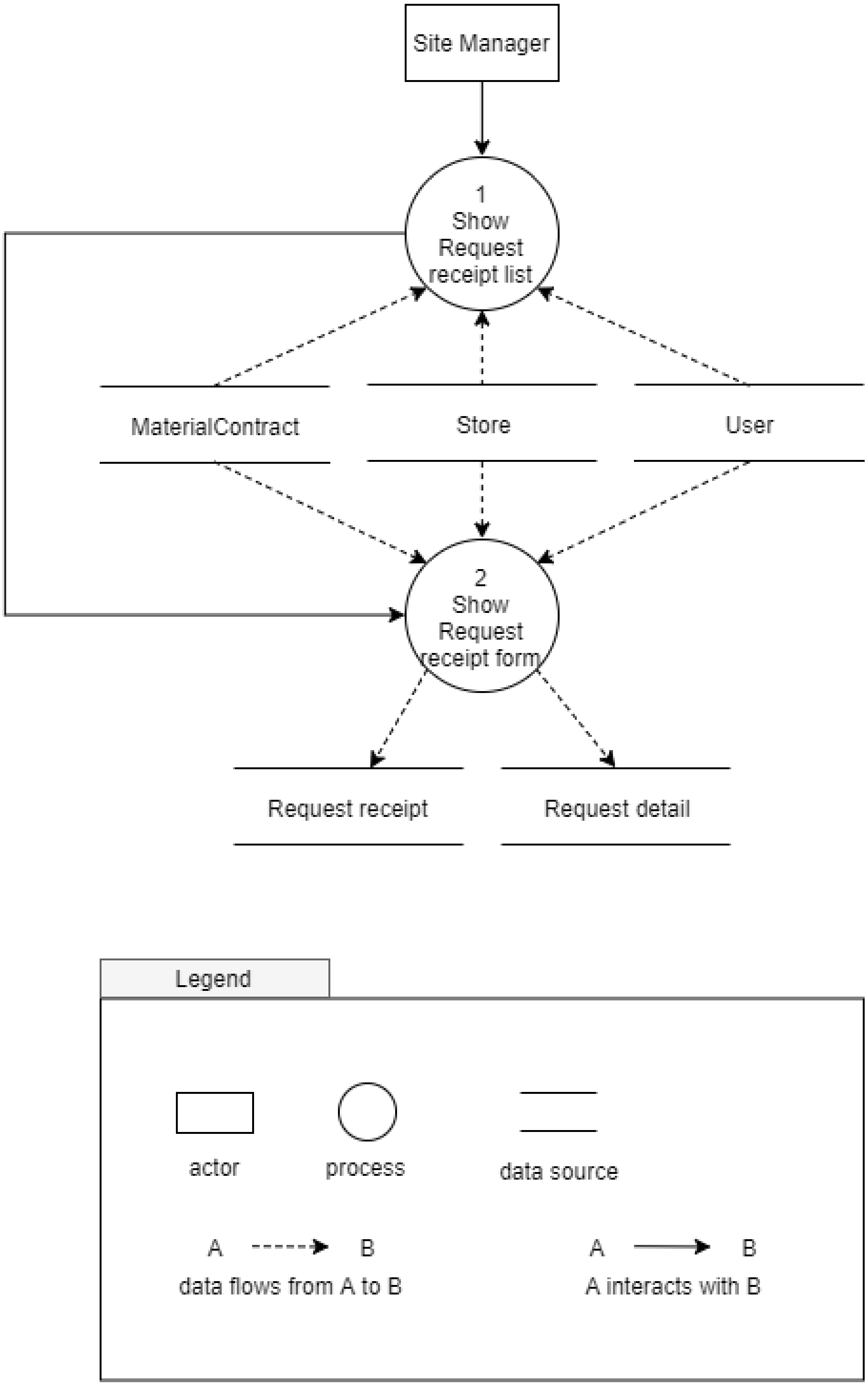

Figure 3.11 Site Manager create Request receipt

3.9.4. SCD announce Stocker to Import Material

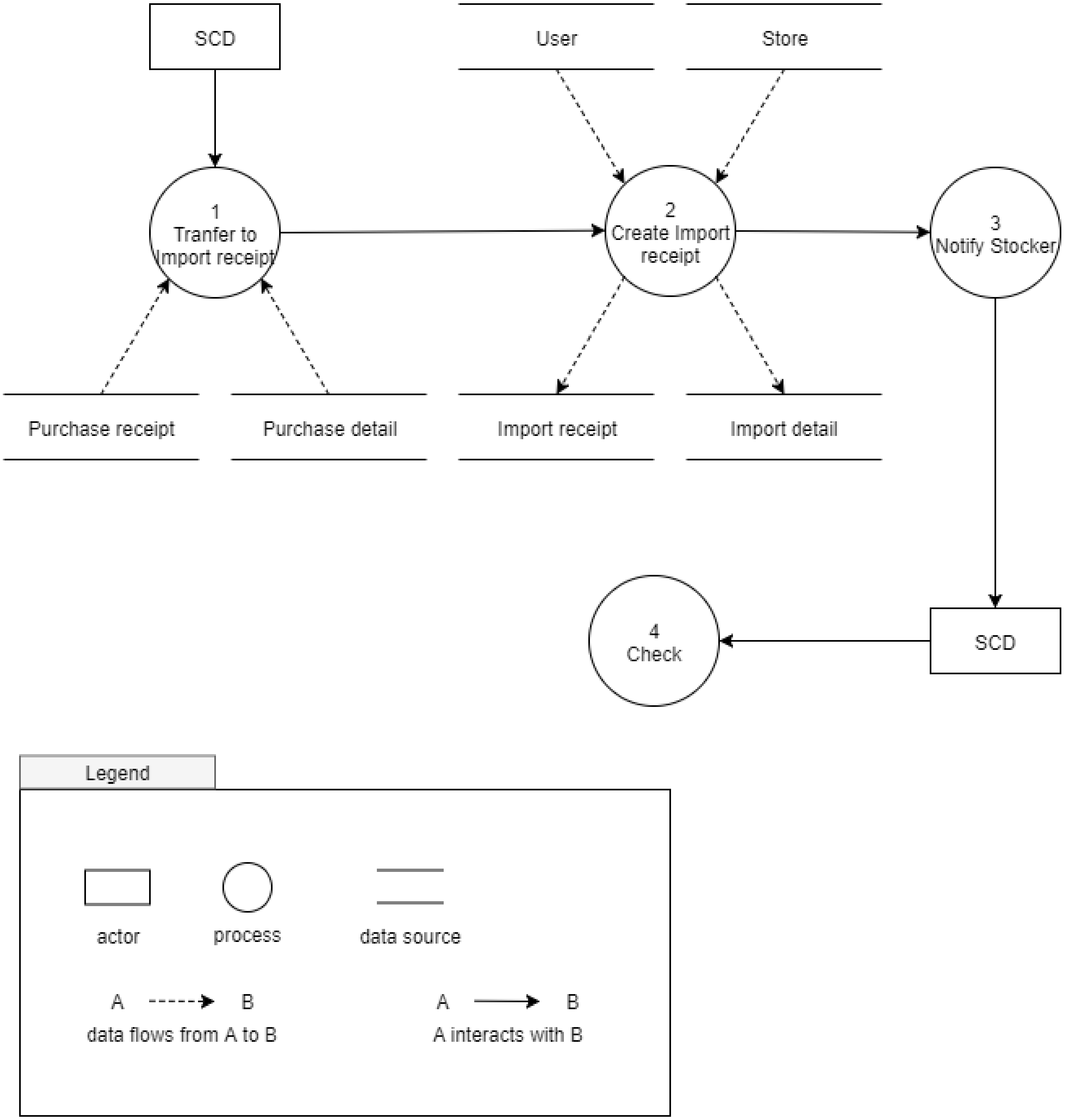

Figure 3.12 C&C View - SCD announce Stocker to import material

3.10. MODULE VIEW

In the Module view, AGEAH designed the view base on Decomposition Style. This style is focusing on the is-part-of relation, which describes the organization of the code as modules and submodules and shows how system responsibilities are partitioned across them. After the Capstone project, RECKLESS team will stop developing this project, so it need to be decompose into small module for others team to continuously develop.

Achievement of certain quality attributes: RECKLESS team uses tactic Reduce size of Module to achieve the Modifiability quality, this means we split module into small pieces and show the relationship of them. It is also easy to maintain or upgrade system component with divide-and-conquer techniques

3.10.1.Project Overview

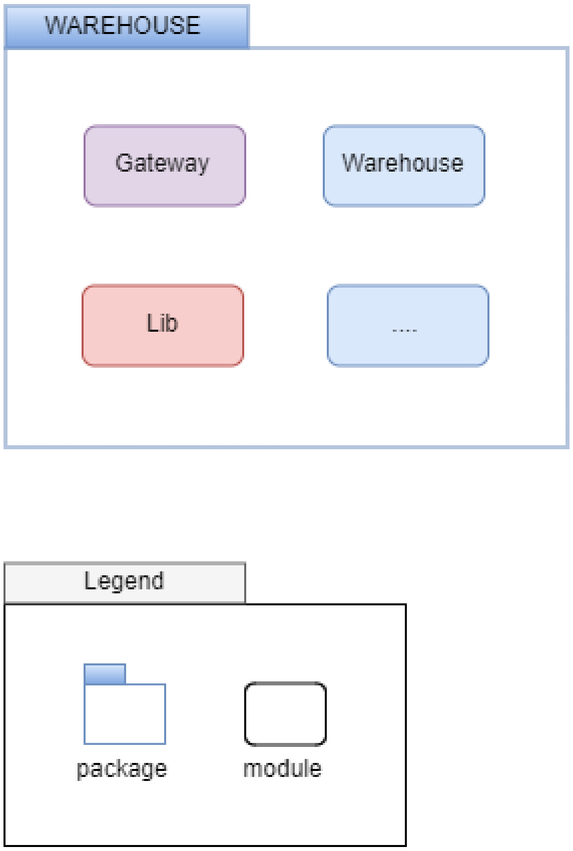

Figure 3.13 Module View - Project Overview

3.10.2.Gateway Module

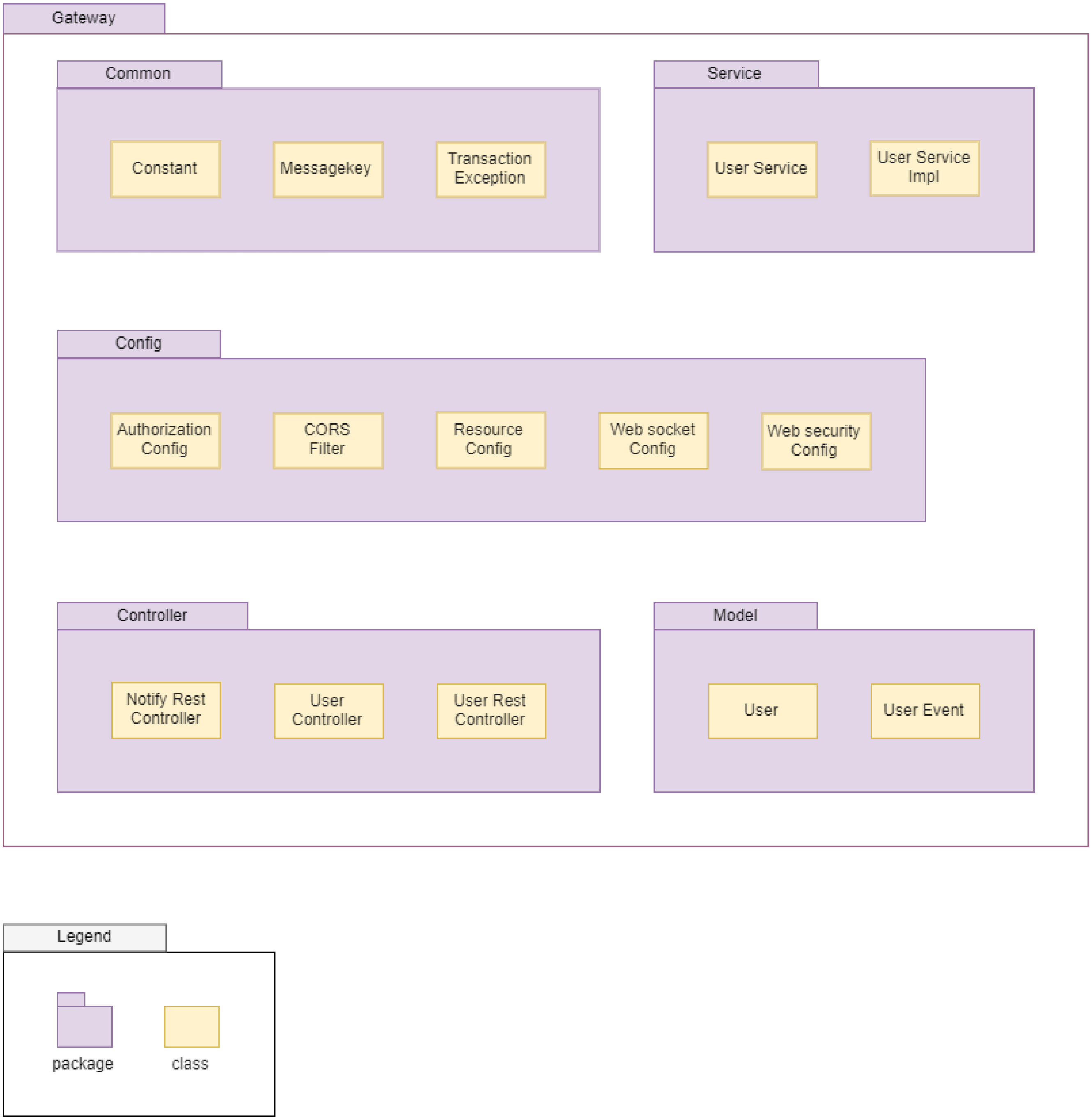

Figure 3.14 Module View - Gateway Module

3.10.3. Warehouse Module

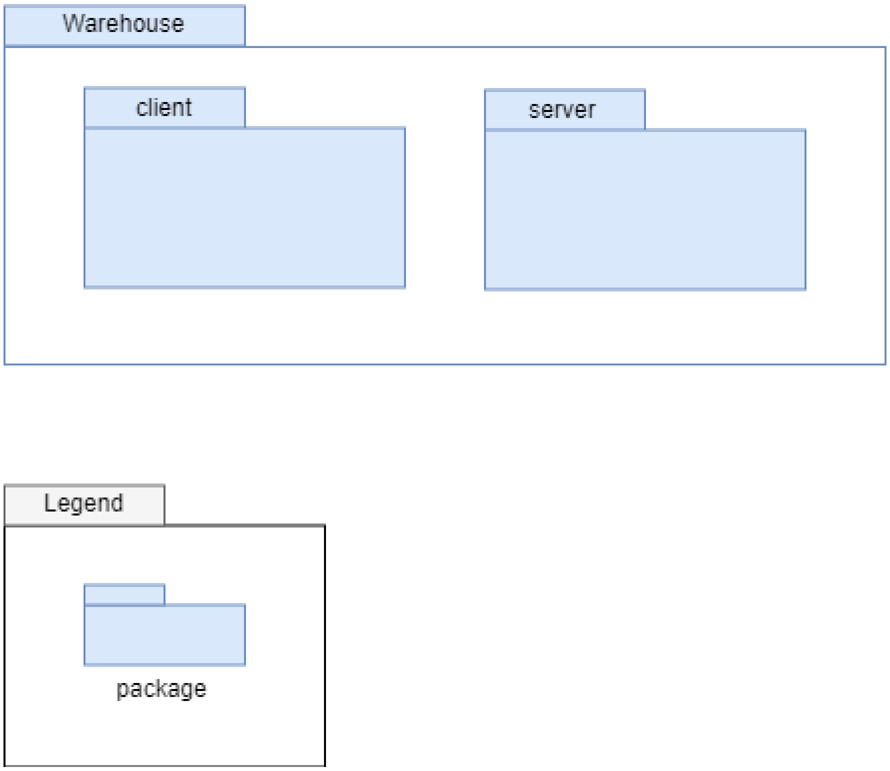

Figure 3.15 Module View - Warehouse Module

3.10.4.Client Warehouse

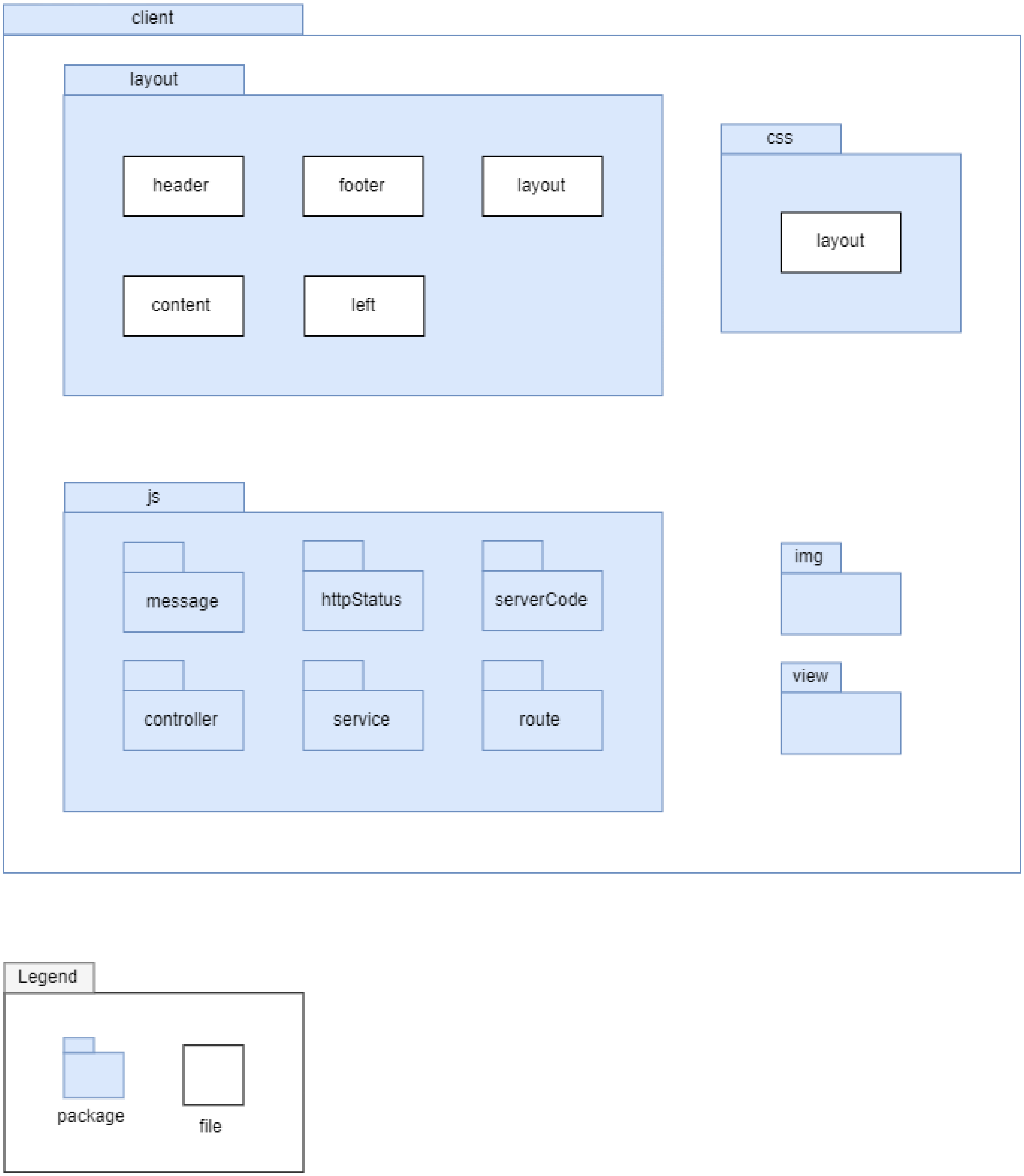

Figure 3.16 Module View - Client Warehouse

Table 14.12 Element of Client Warehouse

Element	Type	Description
Header	file	Content header of the website
Footer	file	Content footer of the website
Layout – layout	file	Content main view of the website
Content	file	Content view of the website
Left	file	Content menu of the website
Layout – css	file	Content the style of the website
Message	Package	Content constant, language translation
httpStatus	Package	Content code
serverCode	Package	Content error code
Controller	Package	Content angular controller
Service	Package	Content angular service
Route	Package	Content angular route
Img	Package	Content image
View	Package	Content view

3.10.5.Server warehouse

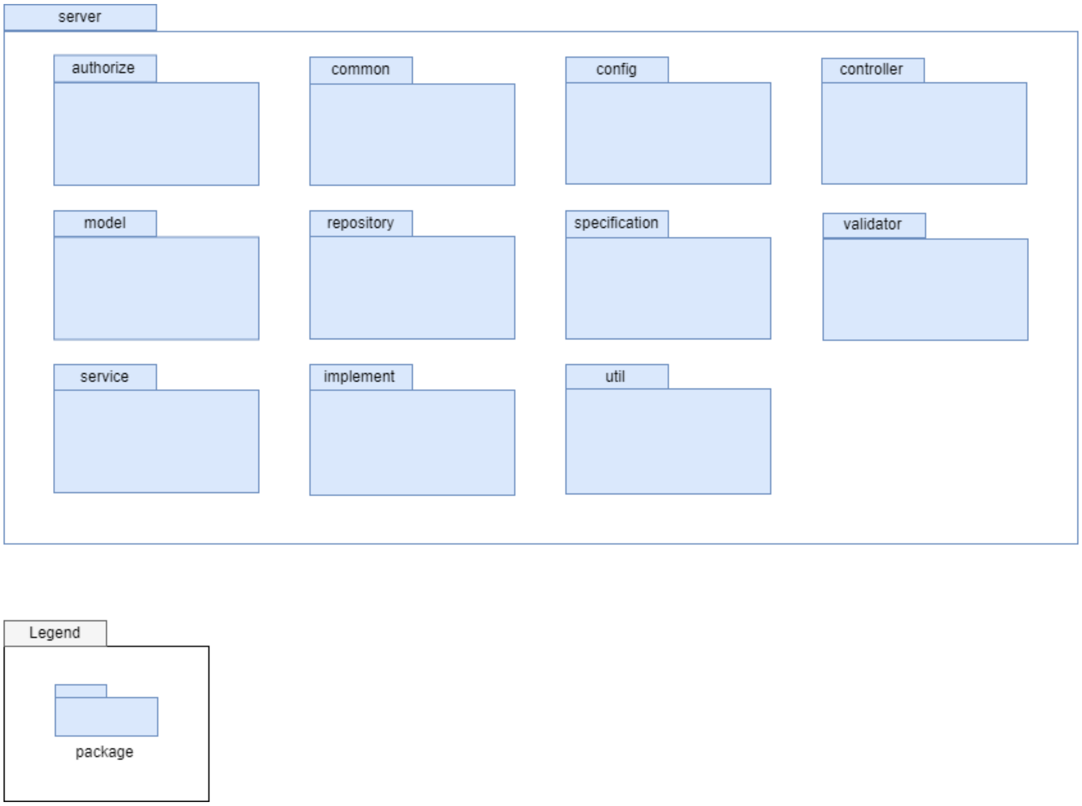

Figure 3.17 Module View - Server warehouse

3.10.6.Library

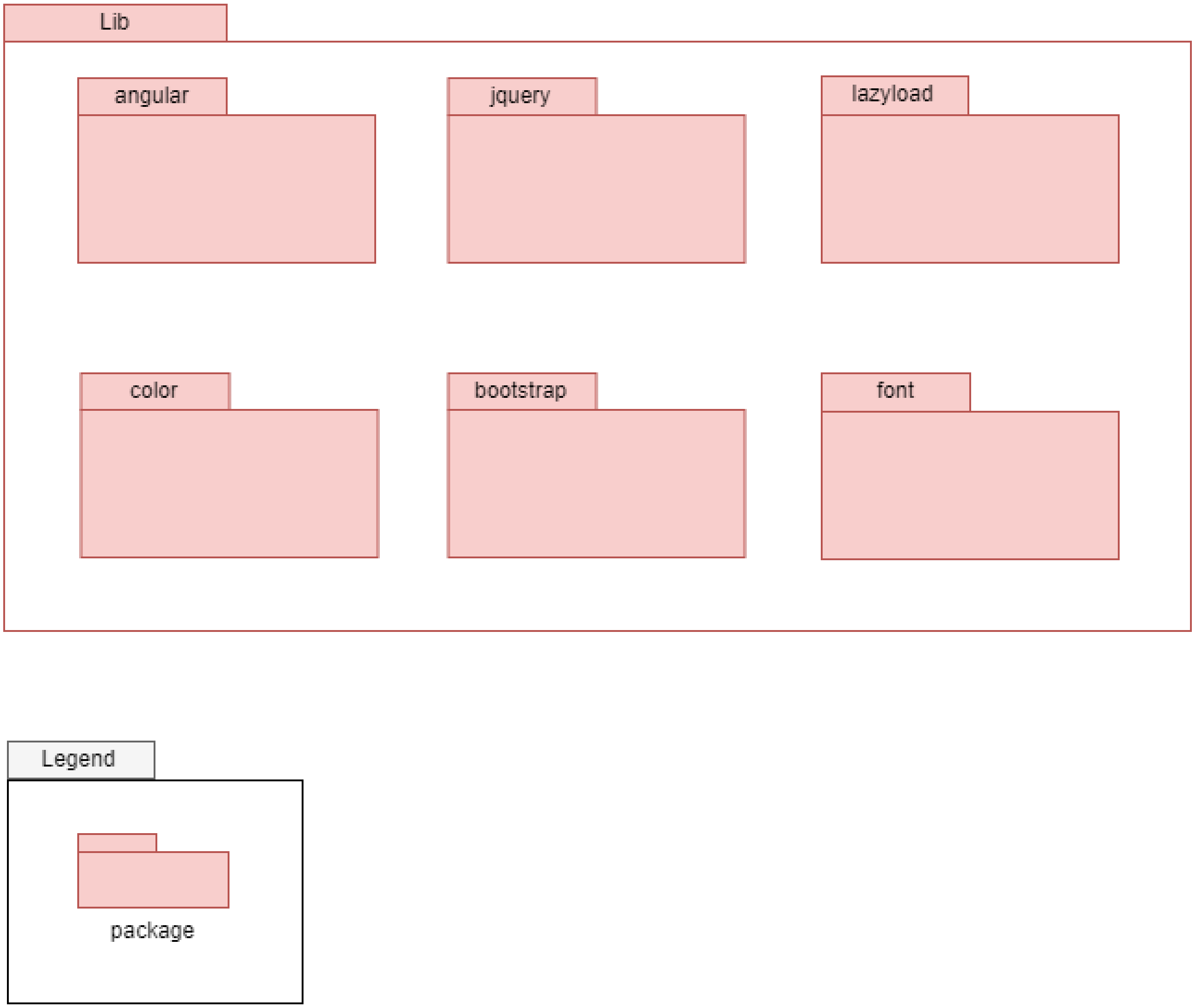

Figure 3.18 Module View - Library

3.11. MAPPING VIEW

3.11.1.Mapping Allocation View and C&C View

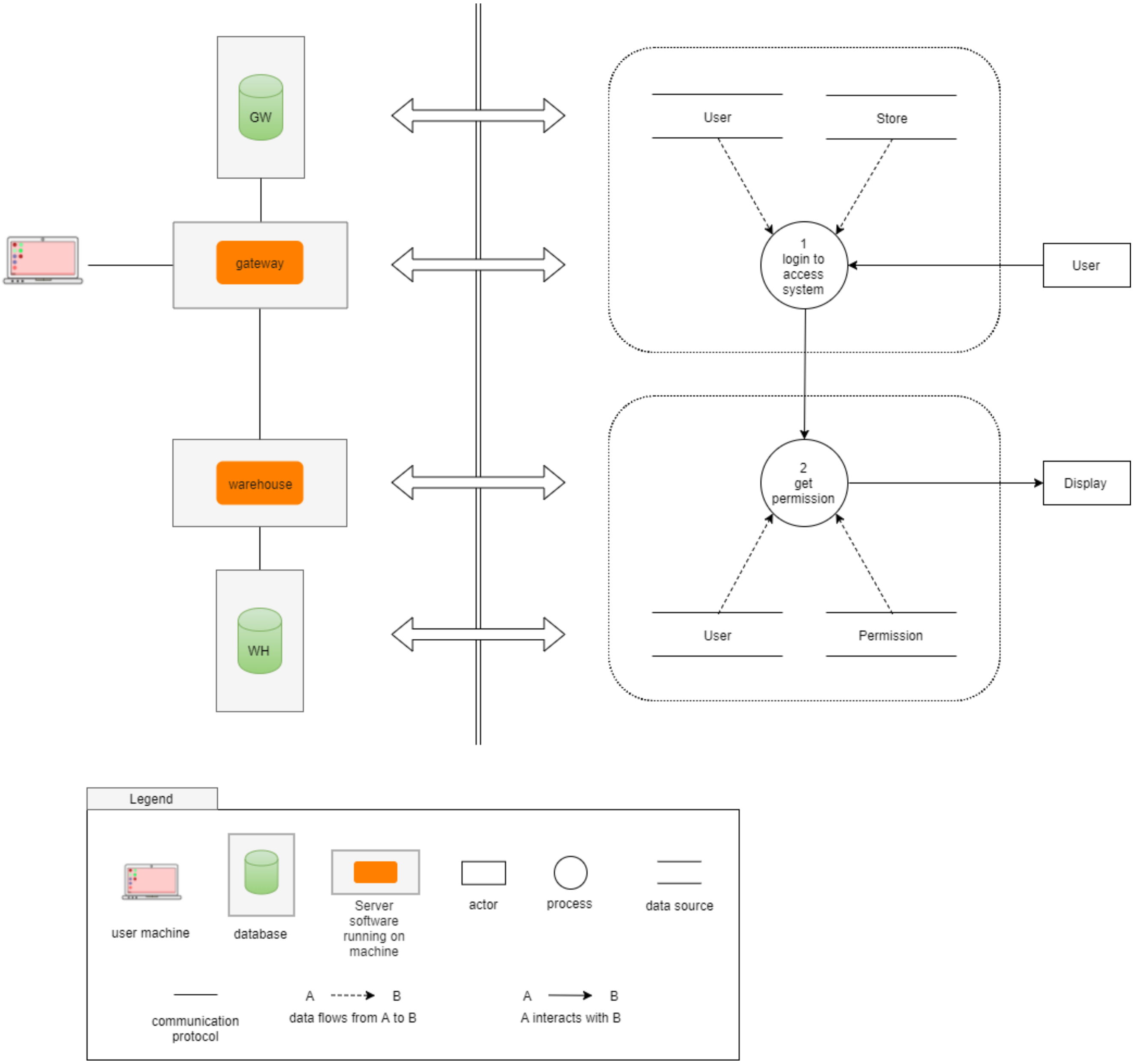

Figure 3.19 Mapping Allocation View and C&C View

3.11.2.Mapping C&C View and Module View

- Gateway

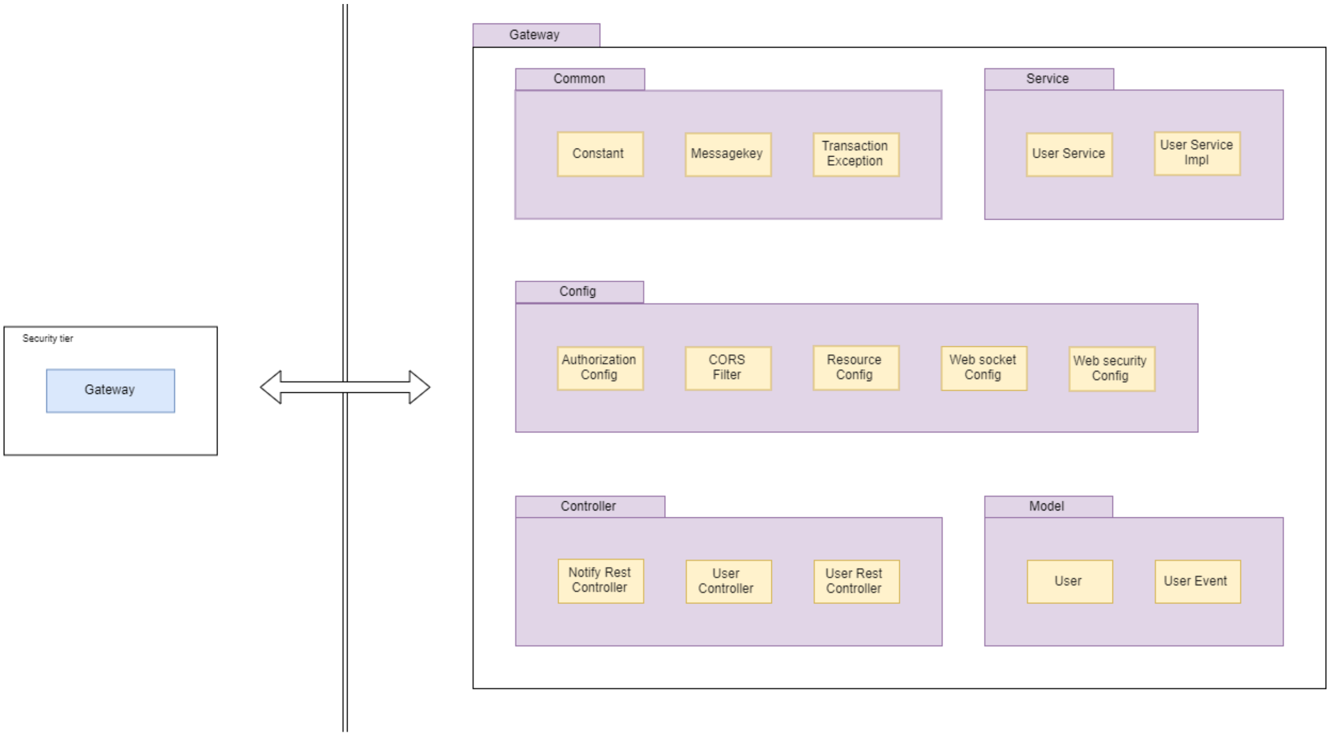

Figure 3.20 Mapping Gateway

- Client Warehouse

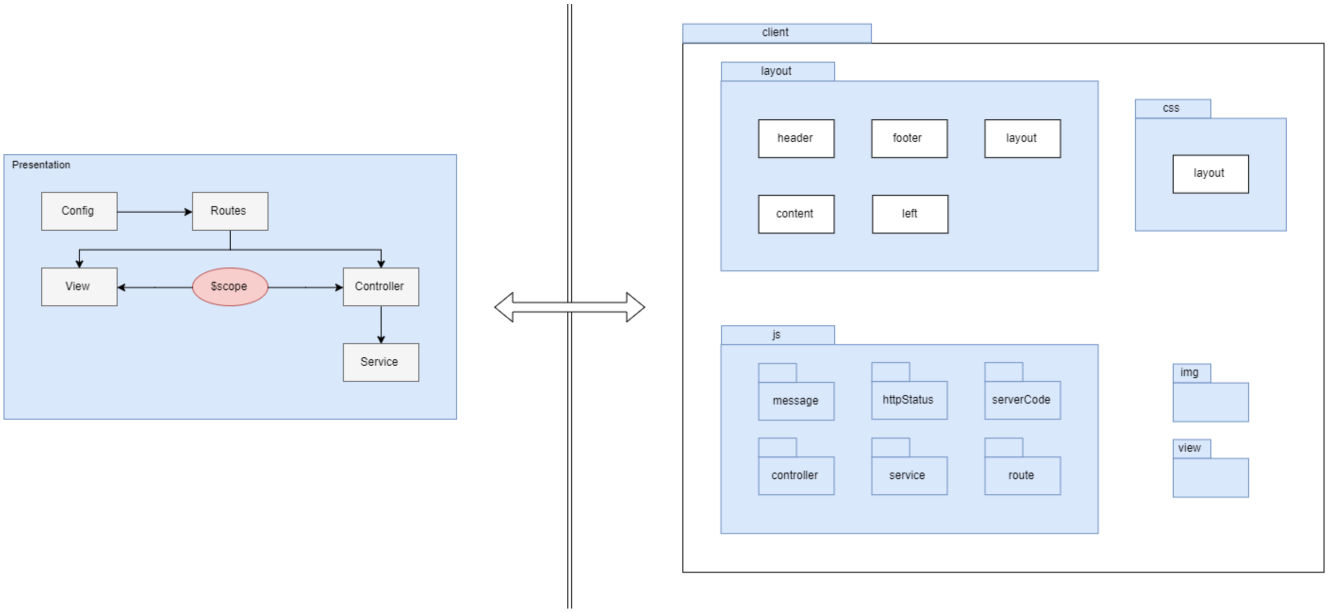

Figure 3.21 Mapping Client Warehouse

- Server warehouse

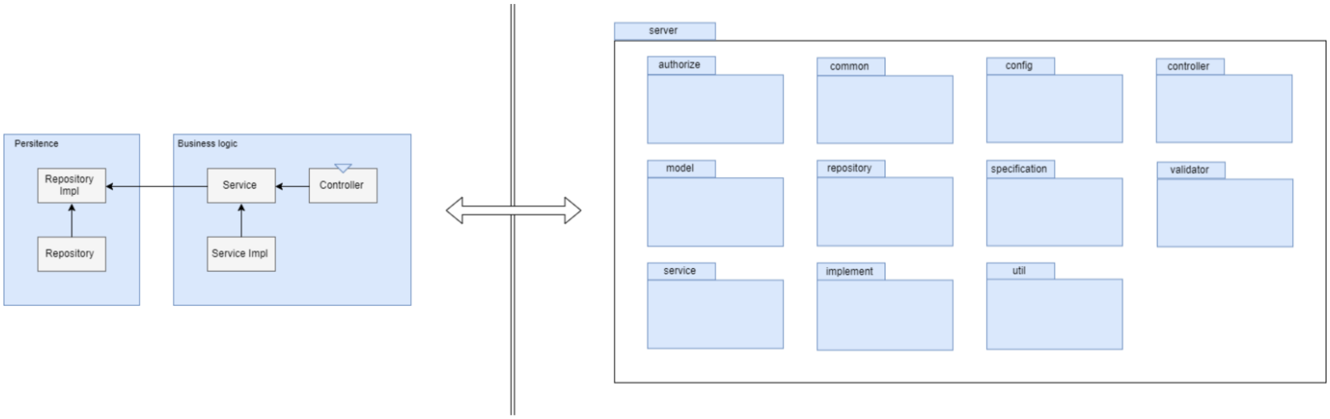

Figure 3.22 Mapping Server Warehouse