SOEN 390

Software Architecture Document

Enterprise Resource Planning

Version 1.0

TEAM 11

03/02/2021

Revision History

Version	Description of Versions / Changes	Responsible Party	Date
1.0	Initial version Add definition of the contents Add use-case, sequence diagrams	Kevin.L Shijun.D Kimchheng.H Saebom.S	Jan.27.2021
1.1	Add detailed view points and correspond diagrams(component and deployment diagrams)	Kevin.L Shijun.D Kimchheng.H Saebom.S	Jan.30.2021
1.2	Modify the version and finalize version 1.1	Kevin.L Shijun.D Kimchheng.H Saebom.S	Feb.2.2021

Table of Contents

1.0 Introduction **1**
- 1.1 Purpose 1
- 1.2 Scope 2
- 1.3 Definitions, Acronyms, Abbreviations 2
- 1.4 References 2
- 1.5 Overview 2

2.0 Architecture Representation **3**

3.0 Architecture Goals and Constraints **5**

4.0 Stakeholders and Concerns **6**

5.0 Viewpoints and Views **9**
- 5.1 Logical View 9
- 5.2 Process View 10
  - 5.2.1 Authentication 11
  - 5.2.2 Signup 12
  - 5.2.3 Product Access 13
  - 5.2.4 CRUD Material list 14
- 5.3 Implementation View 14
  - 5.3.1 Purpose of the Implementation View 14
  - 5.3.2 Overall Implementation 15
- 5.4 Deployment View 16
- 5.5 Use Case View 17
  - 5.5.1 Authentication 17
  - 5.5.2 Product Access 17
  - 5.5.3 CRUD Material list 18

6.0 Architecture Decisions and Rationales **20**

Appendix I Domain model diagram **26**

List of Figures

Figure 1. The “4+1” view model 1

Figure 2. The logical view for the ERP system 9

Figure 3. The process view of the login functionality 11

Figure 4. The process view for the signup functionality 12

Figure 5 Product access sequence diagram 13

Figure 6 sequence diagram of CRUD operations for material list 14

Figure 7. Component diagram of overall ERP system. 15

Figure 8. The deployment view for the ERP system 16

Figure 9. The use case diagram for Authentication functionality 17

Figure 10. Product access use case diagram 18

Figure 11. Use case diagram of CRUD operations for material list 19

Figure 12. Domain model diagram 26

List of Tables

Table 1. Stakeholders and Concerns 6

Table 2. Architecture decision 1 - Overall architecture 21

Table 3. Architecture decision 2- MVC design pattern, front-end language and back-end language 22

Table 4. Architecture decision 3- MySQL database and JPA 23

Table 5. Architecture decision 4- Spring Security and JWT 24

Table 5. Architecture decision 5 - Testing 25

1.0 Introduction

The document presented is intended to illustrate a high level overview of the enterprise resource planning (ERP) system of a company that specializes in the manufacturing of road and mountain bikes.

1.1 Purpose

The Software Architecture Document (SAD) can be considered to serve as a map of the bike manufacturing ERP system. It provides descriptions and an overview of how the software system is actually structured. This is achieved without going into the detail of the actual code since the SAD illustrates the construct of the system via text explanations, diagrams, and other visual representations. Specifically, the “4+1” view model of software architecture by Philippe Kruchten will be used to accurately depict the system at hand.

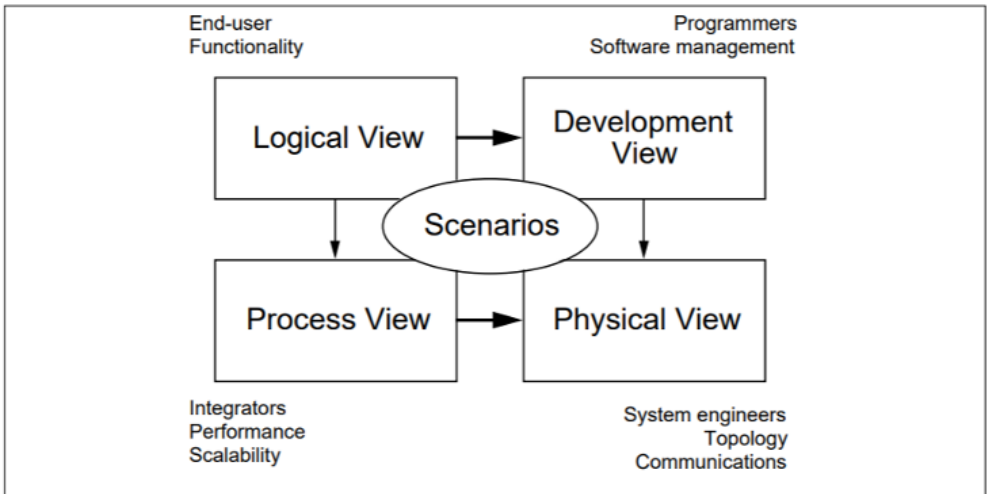

Figure 1. The “4+1” view model[1]

To address and deal with software-intensive systems, the use of multiple and concurrent views, as shown in the figure above, will allow stakeholders to address different concerns regarding the architecture and to handle its functional and non-functional requirements. That is, the description and decisions of the system’s architecture can be organized into five viewpoints, namely the logical view, the process view, the physical view, the development view and the scenarios view.

1.2 Scope

The scope of this SAD itself is to explain the architecture of the bike ERP system. On the other hand, due to time and resource constraints, the scope of the ERP system will be limited to road and mountain bikes. Nevertheless, many more types of bicycles can later be added to the system.

1.3 Definitions, Acronyms, Abbreviations

- **Spring Boot** – application framework and inversion of control container for the java platform, used to build the back end of the ERP web application.
- **React** – an open-source, front end, JavaScript library for building the user interfaces or UI components of the ERP web application.
- **MySQL** – is an open-source relational database management system used for the ERP web application.
- **WWW** – World Wide Web
- **UML** – Unified Modeling Language
- **User** – User who is registered on the ERP site. The user can be a client or admin.
- **SAD** – Software Architecture Document
- **CRUD** – Create, Read, Update, and Delete

1.4 References

[1]. Paper published in IEEE Software 12 (6) November 1995, pp. 42-50
http://web.ist.utl.pt/~fabio.ferreira/material/as/4+1view-architecture.pdf

1.5 Overview

To capture the architecture design and decision or the ERP system, the Software Architecture Document is split into various sections. A description of each of them is shown below.

Section 2: Stakeholders and concerns

Section 3: Architectural goals and constraints

Section 4: Viewpoints and views

Section 5: Consistency and correspondences

2.0 Architecture Representation

This section describes the details of the architecture using the Krutchen's 4+1 Architectural View Model for our ERP system as follow:

Use Case view / Scenarios

Audience: all the stakeholders of the system, mainly for the end-users.

Area: use-case view is a technique for analysing user functional requirements, usually by using the UML to draw use cases. It describes the boundary of the system, and how the Actors related with functional parts of the system to complete their tasks. This view represents the needs of the user in different scenarios and how the system utilizes different services to satisfy users' requirements.

Related Artifacts : Vision Document, Glossary and/or Domain Model, Supplementary Specification.

Logical view

Audience: Analysts, Designers.

Area: It describes functionalities that the system can offer to the end-users. Without describing the relationship with the actors, it focuses on how these functionalities work together within the system boundary. For a large system such as our ERP system, we can represent the logical view for different subsystems and integrate them together. We use UML diagrams such as class diagrams and state diagrams to represent the logical view.

Related Artifacts: Design model which includes class diagrams, interaction diagrams, state diagrams, the subsystems and their interfaces.

Process view

Audience: System Integrators

Area: It describes the dynamic aspects of the system, and focuses on the information flow between different components. Usually, we use sequence diagrams to describe. Besides the data flow, we also need to take concurrency, asynchronization problems into consideration.

Related Artifacts: Process model.

Development/Implementation view

Audience: Programmers

Area: It describes the components of the system in the perspective of programmers. The purpose of this view is software management. Based on it, the programmers can start to code. Usually, we use the UML package diagrams to represent this view.

Related Artifacts: Development model.

Physical/Deployment view

Audience: System Engineers

Area: It describes the hardware requirement or environment for each subsystem, and focuses on the topology of the system components on the hardware layer as well as the hardware connection among them. To represent this view, we use UML deployment diagrams or component diagrams.

Related Artifacts: Deployment model.

3.0 Architecture Goals and Constraints

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

1. One of architectural goals does not only try to understand end-user but also guide a future architecture designer to understand easily about the system. Therefore, the existence of the system can be used as a main nutritious element of building a future advanced system of ERP.
2. The ERPsystem is a web application program which has 3 tier patterns namely presentation layer, domain layer and data source layer. Each layer is driven by following chosen libraries; React, Spring and MySQL respectively. Those 3 tier patterns are packaged up by Docker to make it easier to deploy and run in any environment setup.
3. Since the ERP system contains sensitive data, certain sectors must be encrypted so that the system can protect any data leakage.

4.0 Stakeholders and Concerns

A stakeholder is any individual, group or company that may have any type of direct or indirect relation with, responsibilities towards or interest in the project. Simply put, they are those who may affect or get affected by the outcome of the software project. Below, is a breakdown of stakeholders that have significant contribution in the success of the bicycle ERP system.

Table 1. Stakeholders and Concerns

Name	Description	Concerns
Customers (sales)	Customers are the individuals, groups or companies who purchase the end goods for personal use, re-sell or other.	**Interests:** - Purchasing quality products of desired customization in a timely fashion. **Responsibilities:** - Provide feedback of the end product (bikes).
Vendors and suppliers	They are vendors of raw materials such as metals, rubber and plastic as well as suppliers of parts such as seats, tires, bike frames etc.	**Interests:** - Wholesale or sell in bulk materials and parts needed in the construction of road and mountain bikes. **Responsibilities:** - Provide pricing catered to order size. - Maintain an open communication channel with buyers. - Ensure availability and production of parts and/or raw materials
Transport and shipping companies	These stakeholders constitute those responsible to oversee the transport of goods between vendors, suppliers, users and customers.	**Interests:** - Profit from collaboration with vendors, suppliers and ERP system users to provide transportation and shipping services. **Responsibilities:**

		- Ensure timely, organized and appropriate shipping and transportation of goods. - Take care of shipping logistics, freight policies, pricing etc.
Product manager, system operators, QA inspectors, finance department (users)	These constitute different departments within the bike production company which ensure proper delivery of product and functioning of the production line.	**Interests:** - Work within the bike production company to make sales and ensure fluid operation and ERP. **Responsibilities:** - Ensure production of appropriate bikes. - Ensure quality of the produced bikes. - Enable operations by keeping tabs and logs of payments required.
Factory workers and builders	Members of the assembly line of the bicycles.	**Responsibilities:** - Prepare, and build bikes as per quota.
Project manager, architect, developers, tester, maintainer	This category of stakeholders mainly encompasses those who work hands on or closely with the ERP system that serves the production company.	**Interests:** - Provide ERP solutions. - Profit from companies using their ERP system. **Responsibilities:** - Ensure market demand. - Project funding (i.e. homegrown solutions), setting budgets, assigning developers. - Maintain product and monitor its progress. - Communicate with other stakeholders for use cases & system requirements. - Ensure product evolution and flexibility - connectivity etc.) to provide accurate and optimized devices.

Network, service and platform providers	These are the stakeholders that provide connectivity and allow communication to be integrated between devices and machinery.	**Responsibilities:** - Supply and handle network infrastructure and provide reliable connection between devices. - Provide hardware to support or enable connected services (i.e. internet connection) - Allow communication between devices via Bluetooth, NFC or LAN microchips.
Regulators	These may include policy makers, stakeholders who ensure standard of operation and domain experts in the government.	**Responsibilities & interests:** - Overseeing and provision of regulatory framework with appropriate targets and subsidies which gives a direction to entrepreneurialism and innovation towards carbon imprint/emission reduction and energy efficiency solutions. - Address ethical issues of technology. - Address working conditions of the assembly line.
Negative users	This category include hackers and other malicious users	**Concerns:** - Does the ERP system protect all forms of unauthorized access? - Is appropriate access given to the wrong type of users? - Is data properly encrypted?

5.0 Viewpoints and Views

5.1 Logical View

The logical view's main concern is with the functionality (i.e. functional requirements) that the ERP system provides to end-users. Here, the end-users are those who operate the ERP system within the bicycle manufacturing company.

More specifically, this view takes the problem domain, and decomposes it (the ERP system) via abstraction into objects or object classes. This allows for not only the functional analysis but also, to identify the design elements across various parts of the system. Typically the use of various types of representation such as UML, class or state diagrams are employed with an object-oriented (OO) style. The following diagram tries to illustrate a simple description of the system behavior while avoiding premature specialization of classes that are yet to be determined in the following sprints of the ERP system.

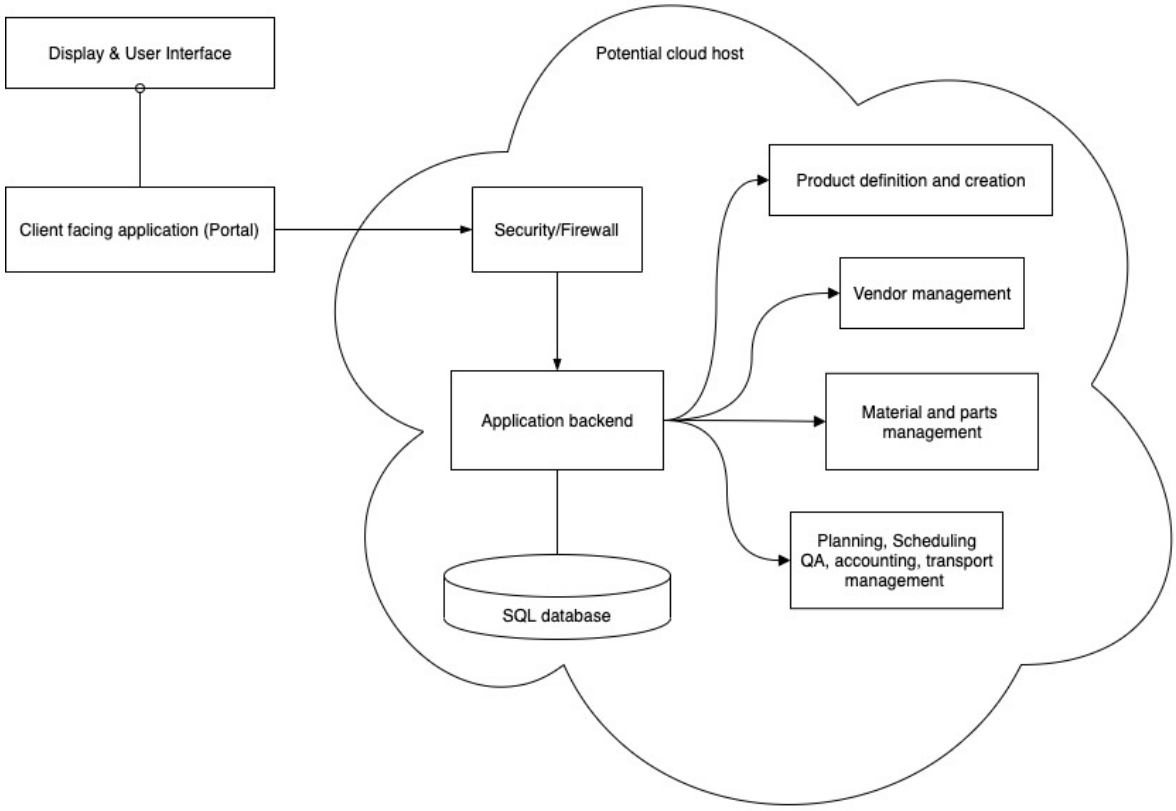

Figure 2. The logical view for the ERP system

- **Client facing application:** As either a web or mobile application, the portal utilizes display and UI to allow interaction with its users. The application consumes the ERP system's API.

- **Security/Firewall:** Protects the system against unwanted operations and network attacks. Users are given different access according to their title.

- **Application backend:** Business logic that allows for the creation of a bike all the way to it being shipped out to consumers. This includes multiple sub-categories of management systems separated and abstracted according to use cases.

- **Potential cloud host:** Possible service to use to host the bicycle ERP system in future iterations.

- **Database:** The database utilized for the ERP system is mySQL.

5.2 Process View

In this section, the use-case and process views are shown. The use cases are a set of scenarios and actions experienced and performed by users that represent some significant central functionality of the ERP system. Currently, the SAD covers the login, product access, and material list manipulation use cases.

The process view focuses on the dynamic aspects of a given system. Here, it explains the run-time behavior of the bike ERP system and addresses issues such as concurrency. For each use case covered in the use-case view, there is a corresponding sequence diagram in the process view which illustrates the sequence of control, concurrency and passing of data within the ERP system.

5.2.1 Authentication

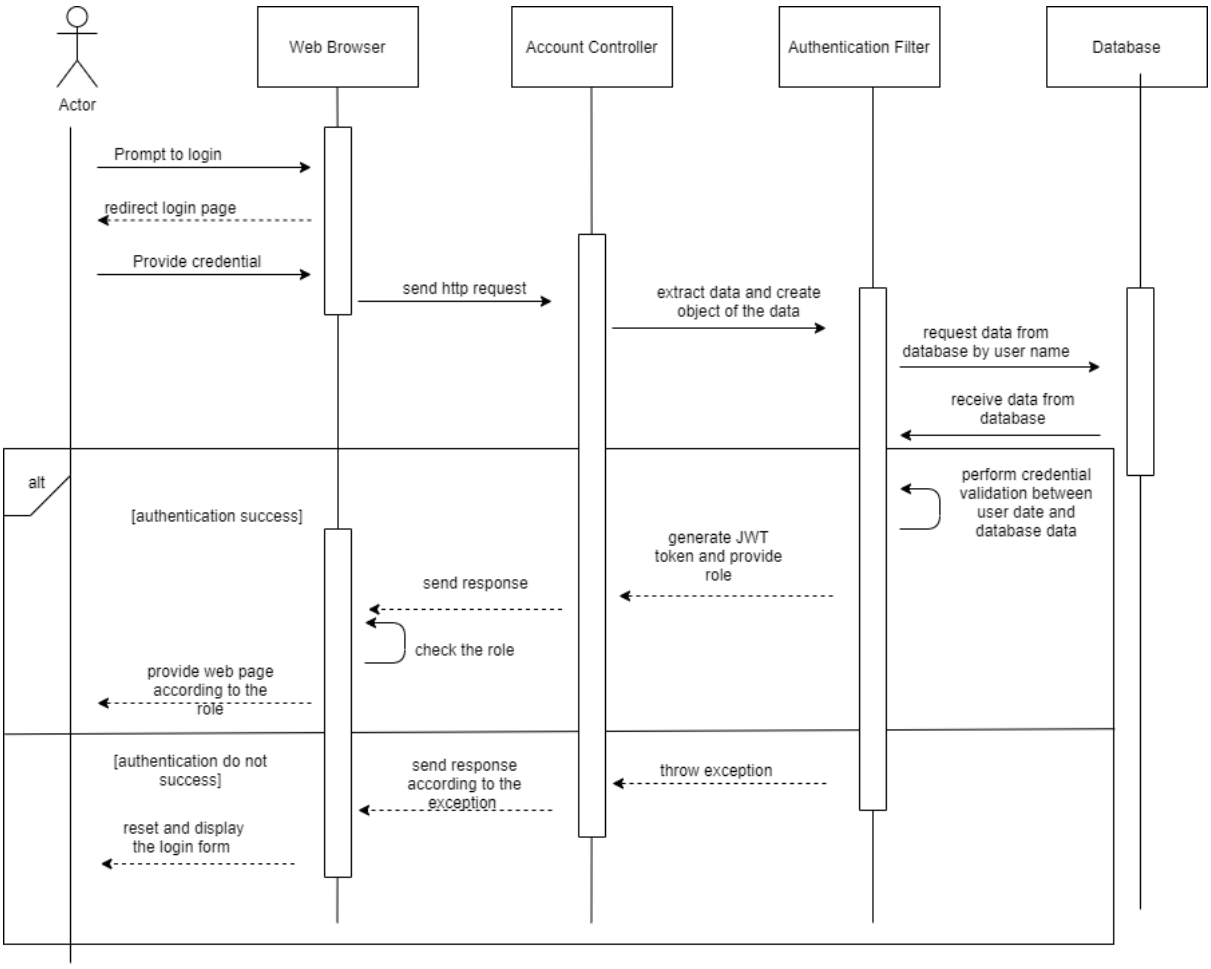

Figure 3. The process view of the login functionality

5.2.2 Signup

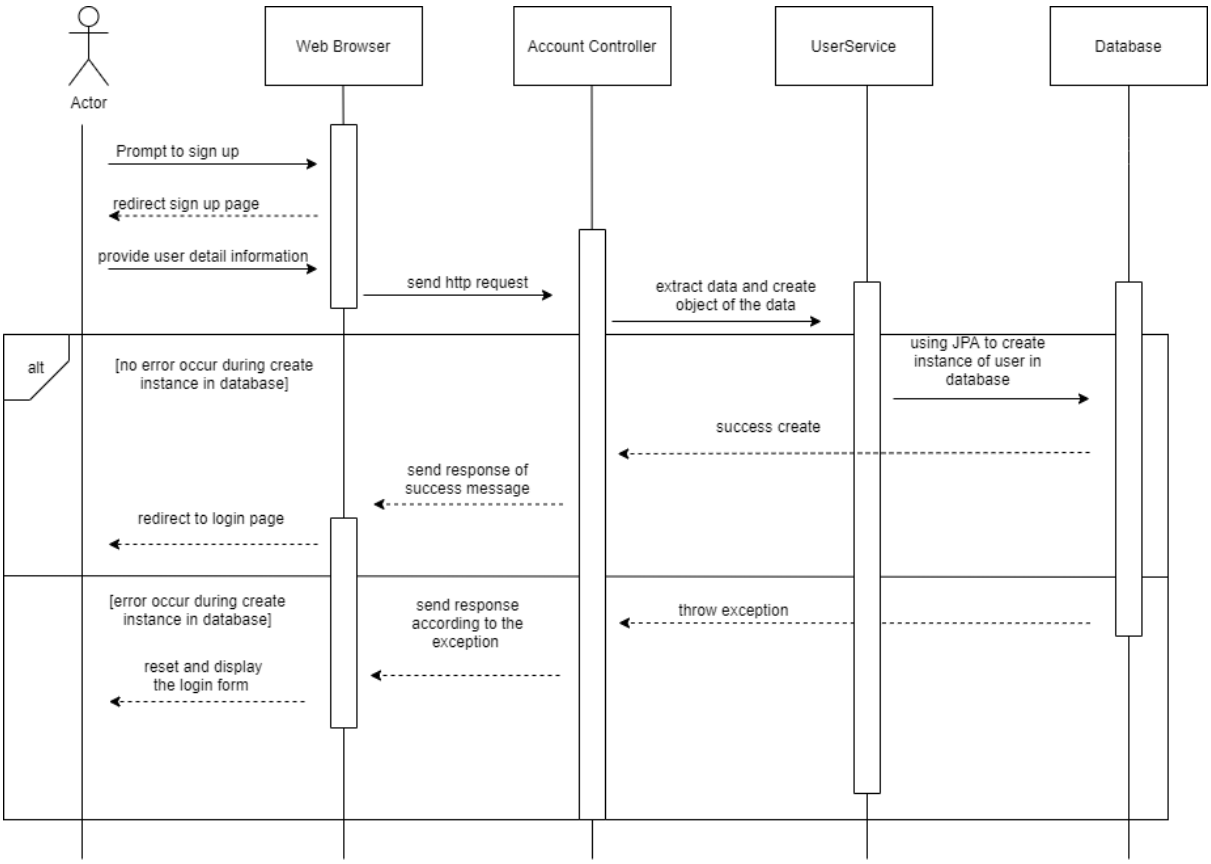

Figure 4. The process view for the signup functionality

5.2.3 Product Access

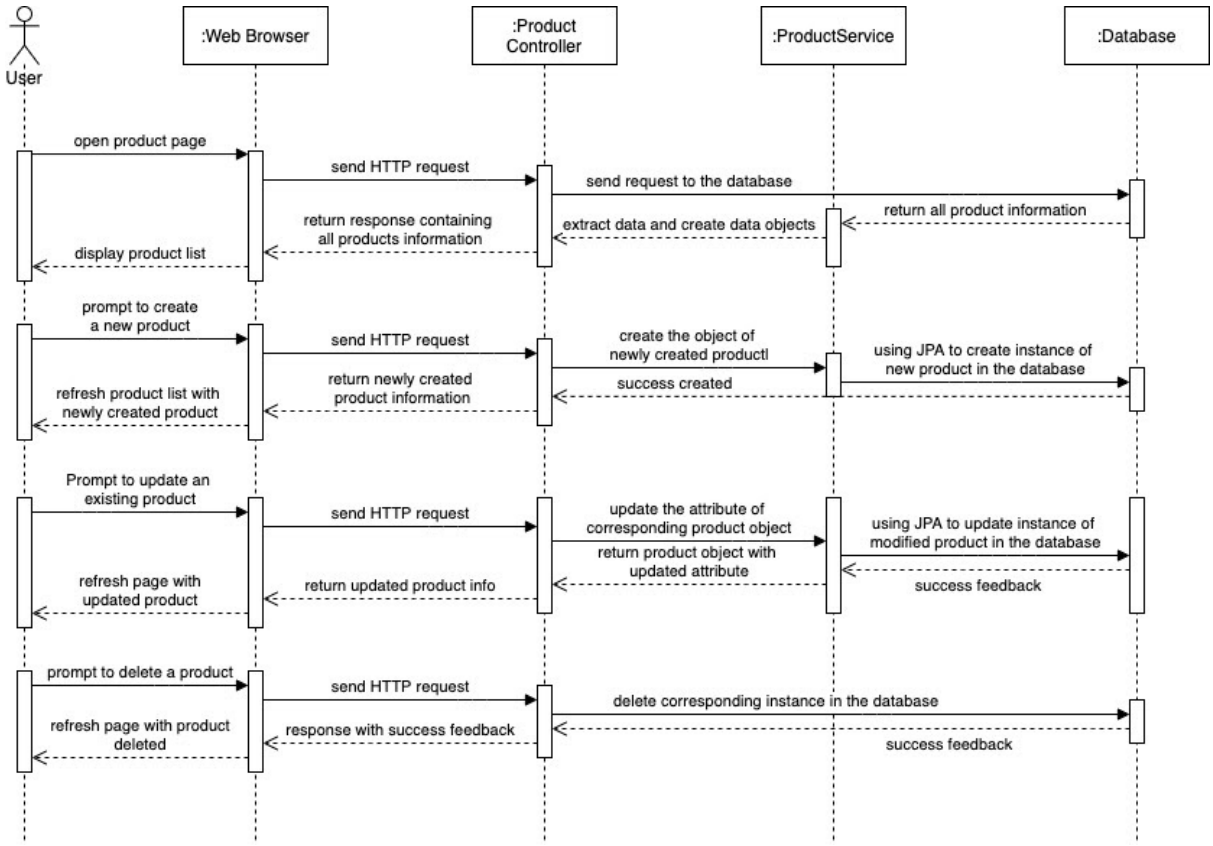

Figure 5 Product access sequence diagram

5.2.4 CRUD Material list

CRUD material list has the following functionalities:

- Displaying the latest material list to the user.
- Creating a new material, save the corresponding information into the database and display the updated material list to the user.
- Updating the material information to the database and display the updated material list to the user
- Deleting a material from the list and the database, and display the updated material list to the user

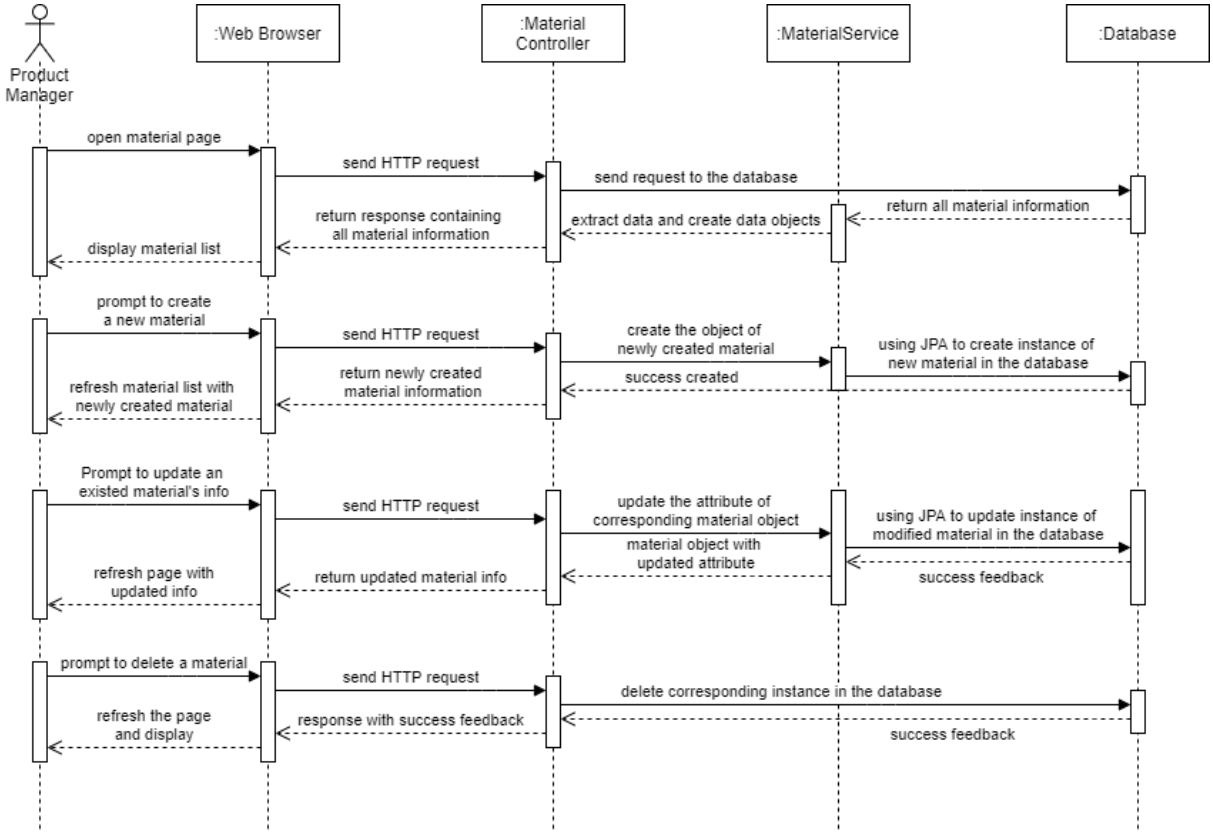

Figure 6 sequence diagram of CRUD operations for material list

5.3 Implementation View

5.3.1 Purpose of the Implementation View

Purpose: The view is to capture the architectural decisions made for the implementation so that it helps individuals, teams or contractors to understand easily about the behavior of the system. Even further, the implementation view makes it easy to assess

the amount of code to be developed, modified, or deleted(scaling the code usage) .
Typically, the implementation view contains:

- Component diagrams to illustrate how subsystems are organized in layers and hierarchies.
- Describe or illustrate dependencies between subsystems.
- Enumerate all subsystems through the model.

5.3.2 Overall Implementation

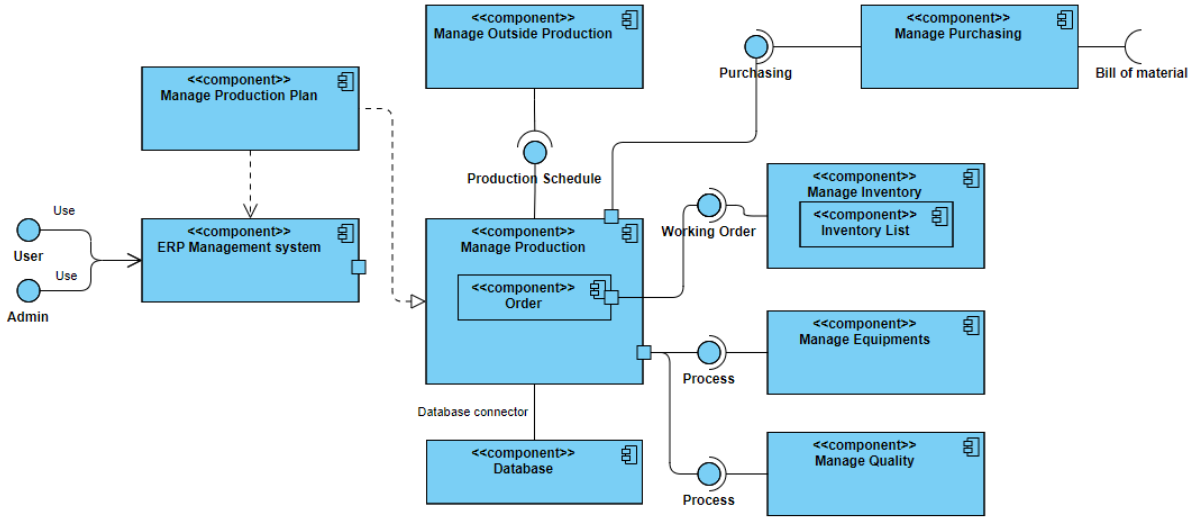

Figure 7. Component diagram of overall ERP system.

To adopt the patterns that describe the relationship between the system and subsystems, the component diagram has been introduced, as looked in Figure 7, it includes 2 tiers. All subsystems are derived by production management, which is also a subsystem itself, comes from the ERP management system, to conjoint with other subsystems such as inventory, quality, and purchasing management system. Additionally, all data is stored in database so that users can manage large amounts of data efficiently and accurately.

5.4 Deployment View

5.4.1 Purpose of Deployment View

- Display the components of the hardware and how these components communicate with each other.
- Project the software system into the hardware configuration
- Help to design the software architecture
- Document the deployment of the software for later usage

5.4.2 Overall Deployment

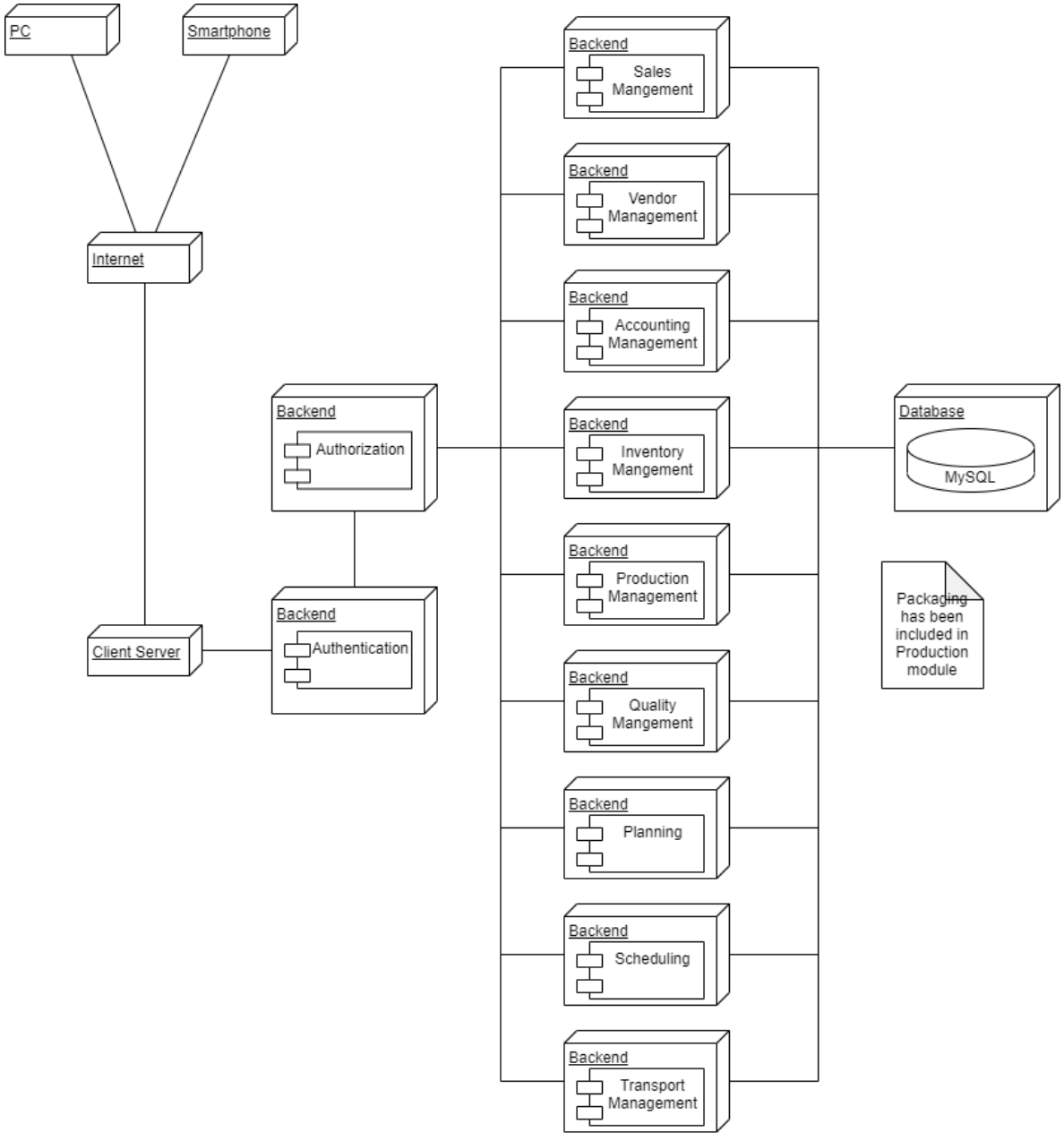

Figure 8. The deployment view for the ERP system

5.5 Use Case View

5.5.1 Authentication

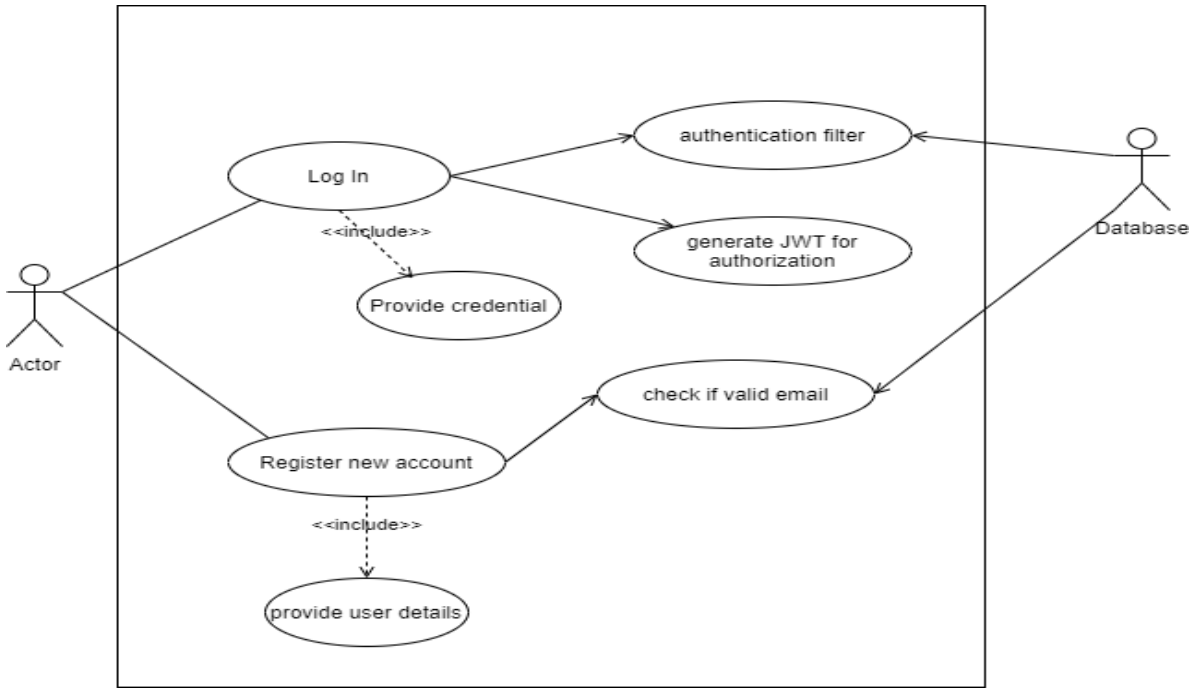

Figure 9. The use case diagram for Authentication functionality

5.5.2 Product Access

- **Access products:** Registered users can access the catalog and detail of available products.
- **Search product list:** Any user can search the catalog of available products regardless of title.
- **Create new product:** Administrators can create new products and add it to the product list/catalog.

  Delete existing product: Administrators can delete existing products and thereby initiate a chain of command to remove production of a given model and dispose or transfer raw material and/or parts.
- **Modify existing product:** Administrators modify materials and parts that go in the making of existing products.
- **Input validation and user authorization:** The system validates user input and verifies them against database and availability (i.e. of raw material or parts). The

system also ensures to give the right access privileges depending on the type of user account.

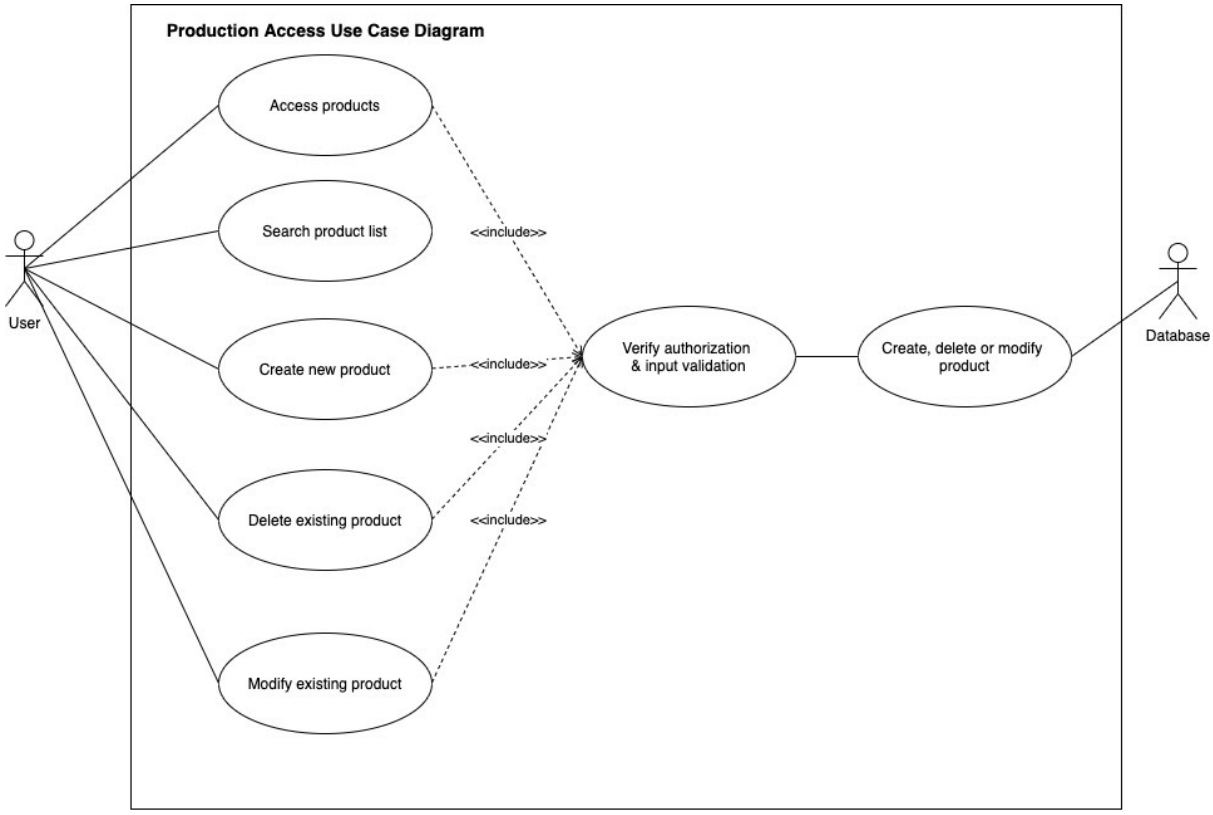

Figure 10. Product Access Use Case Diagram

5.5.3 CRUD Material list

The Admin is allowed to do the CRUD operations to the material list. From the requirements, the system should be able to:

- Authorized the product manager to operate the material list and prevent unauthorized users to access these functionalities
- Authorized users should be able to create, read, update and delete material list
- Create, Update and Delete operations should be able to modify the corresponding data in the database, and the displayed material list should be refreshed after the operations have been done so that the users can have the correct record.
- Validate the inputs for creating and updating the material list.

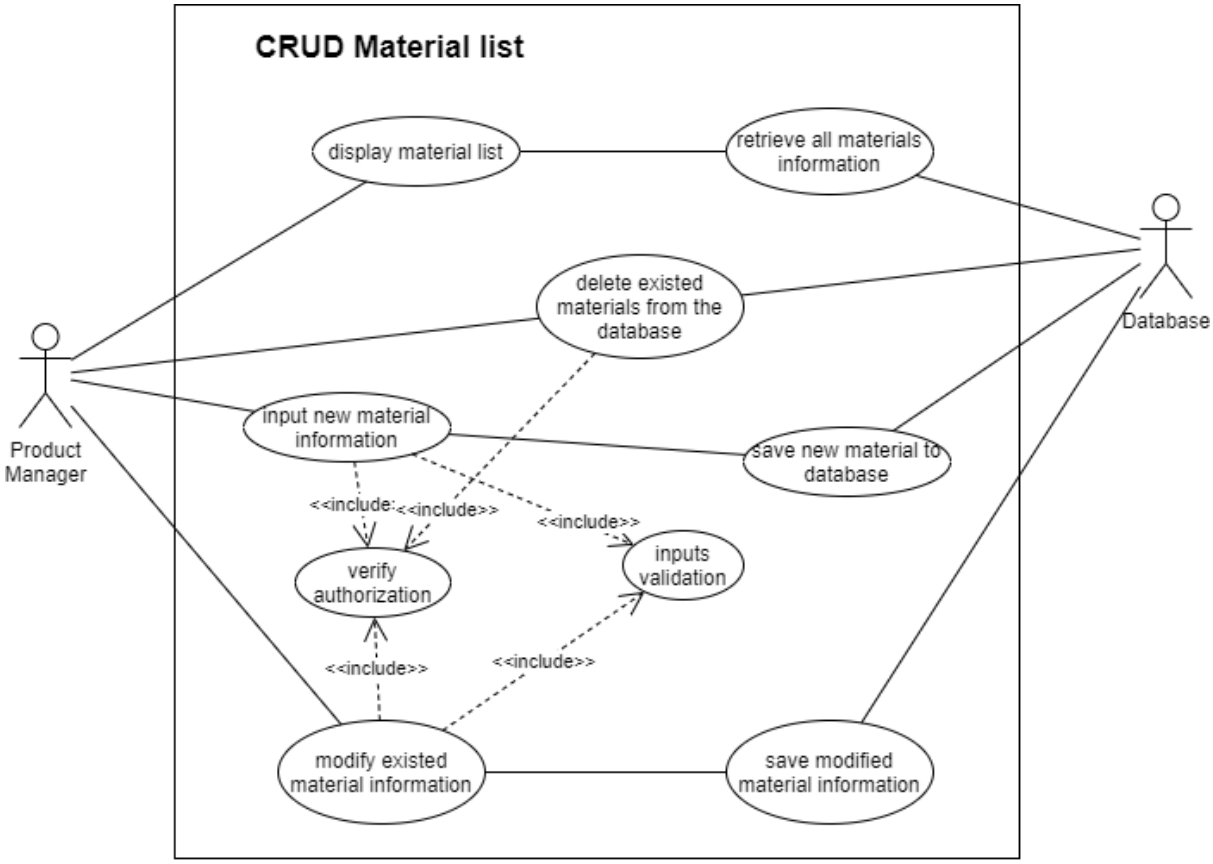

Figure 11. Use case diagram of CRUD operations for material list

6.0 Architecture Decisions and Rationales

Table 2. Architecture decision 1 - Overall architecture

Related Requirement	The system shall be mobile friendly web.
Alternative	Mobile Application(iOS and android), Desktop application(JavaFx, C# etc)
Constraints	Online web application, rely on the internet to communicate between client and server.
Assumption	All devices have the internet access and installed browser.
Architecture Decision	
Identifier	#1
Description	ERP solution will be a web application.
Rationale	● The client does not have to install application ● Responsive web page access for all devices

Table 3. Architecture decision 2- MVC design pattern, front-end language and back-end language

Related Requirement	The system shall follow the MVC design pattern.
Alternative	MVVM(Model view view model), MVP(Model View Presenter)
Constraints	The developer must be familiar with MVC design patterns. The developers have basic knowledge about java and javascript.
Assumption	All browsers are javascript enabled and could run ECMAScript 5 or higher.
Architecture Decision	
Identifier	#2
Description	ERP solution will separate front-end from back-end. By using asynchronous Javascript and JSON to exchange the data. ERP would use Java(Spring boot) and Javascript(React).
Rationale	● MVC would make an independent between front end and back end. Therefore, changing or update one side does not affect another side ● SpringBoot is chosen for the back-end. ● Restful API would be used for the controller. ● React is implemented for the view. ● Some models would be represented by a simple java class some would be used as java beans.

Table 4. Architecture decision 3- MySQL database and JPA

Related Requirement	The system must allow defining, ordering, and tracking raw material and final products.
Alternative	NoSQL, JDBC(Java Database Connectivity) and POJO(Plain old java object)
Constraints	The developers have basic knowledge about databases and some web design patterns..
Assumption	The developers have installed needed software like MySQL. There will not be a lot of data(Gigabytes)
Architecture Decision	
Identifier	#3
Description	ERP solution will use MySQL database to store the data, JPA for data mapping and table data pattern for CRUD repository.
Rationale	● MySQL, which is a relational database, helps to access the data faster and efficiently. ● JPA follows the data mapping pattern of enterprise application architecture. JPA helps mapping the java class to a table in the database. ● Table data patterns are used to separate the query to the database from the resource or data itself.

Table 5. Architecture decision 4- Spring Security and JWT

Related Requirement	Sensitive data must be encrypted (credential details, personal information)
Alternative	MD5(message-digest), RSA(Rivest–Shamir–Adleman)
Constraints	Hash function encryption is one way, there is no way to decrypt. The spring security framework has to be used to confirm the encryption.
Assumption	The developer has some basic knowledge about cryptography.
Architecture Decision	
Identifier	#4
Description	ERP solution will use Spring security for encryption the password and authentication.
Rationale	● Spring security using hashing and salting to encrypt the password. Hashing is one way function. Therefore storing the encryption password in the database is secure.

Table 5. Architecture decision 5 - Testing

Related Requirement	The system shall follow the MVC design pattern and at least 50% test coverage for Controllers classes.
Alternative	TestNG , NUnit , openClover
Constraints	Testing could only identify if there is a present of the bug in the software but it cannot prevent the bug.
Assumption	The developer has basic knowledge about testing.
Architecture Decision	
Identifier	#5
Description	ERP solution would use Junit testing and Jacoco.
Rationale	● Junit would be unit testing for the software. ● Jacoco would be used for test coverage of line, method, class etc. ● Could perform Spring testing if it is necessary.

Related Requirement	The system sends e-mails to certain roles
Alternative	
Constraints	The developer has basic knowledge about electronic mail and user agents.
Assumption	The email that is stored in the database is a valid email.
Architecture Decision	
Identifier	#6
Description	ERP solution will use SMTP(Simple Mail Transfer Protocol) to send email.
Rationale	• With help of the spring library, ERP could deliver the message to the receiver server.

Table 5: Architecture decision 6- SMTP

Appendix I Domain model diagram

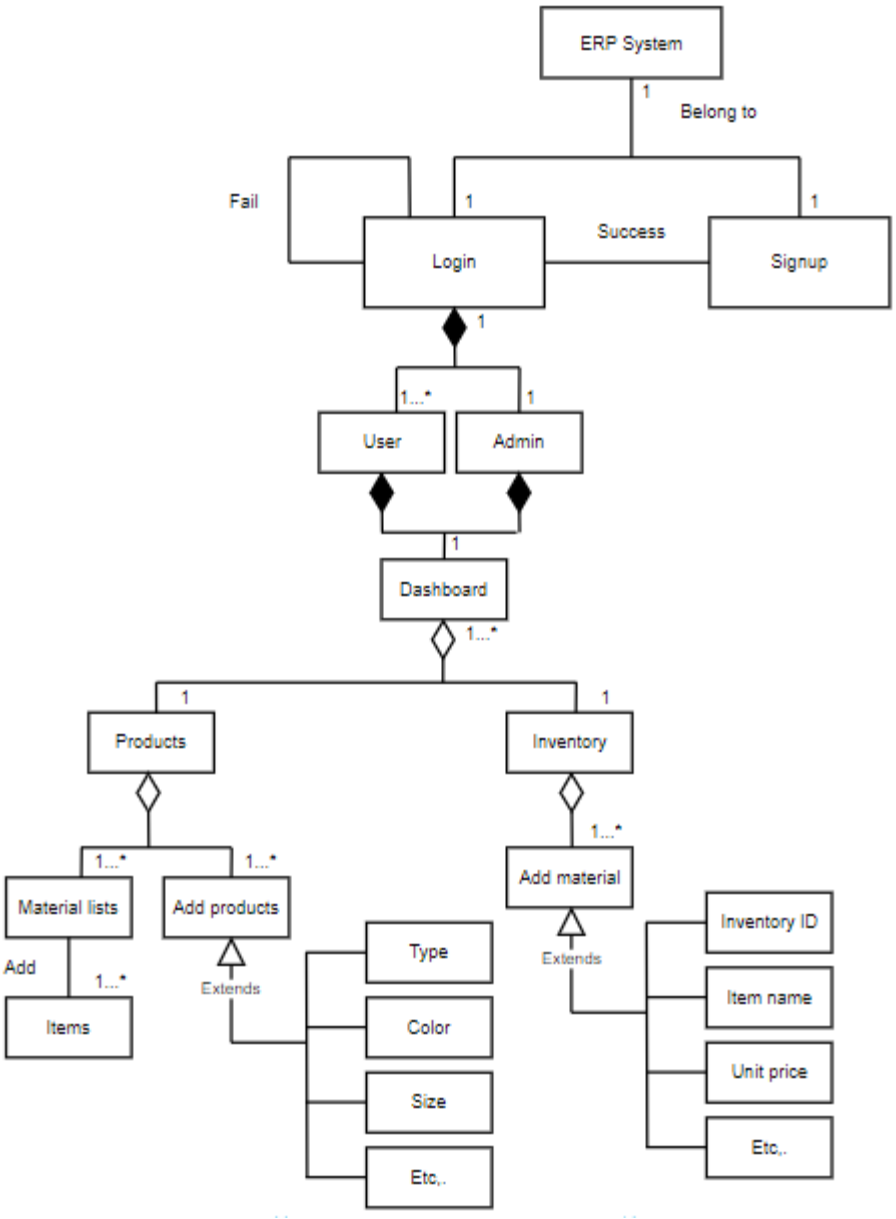

Figure12. Domain model diagram(sprint 1 scope).