

**Inventory Manager Design**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Revision History** | | |  | |  | |  |
|  | **Date** | **Issue** |  | **Description** |  | | **Author** | |  | |
|  |  |  |  |  |  | |  | |  | |
|  | 09/25/2019 | 1.0 |  | Initial version of the document |  | | Tekelmariam | |  | |
|  |  |  |  |  |  | | Roble Tesfamariam | |  | |
|  |  |  |  |  |  | | Filimon | |  | |
|  |  |  |  |  |  | | Luwam Tesfamariam | |  | |

**Table of Contents**

1. Introduction ..............................................................................................................................................4

2. Requirements - Use-Case – Usage Scenarios..........................................................................................4

3. High Level Design.....................................................................................................................................7

4. Detailed Design – sequence Diagrams ...............................................................................................10

5. Issues ...............................................................................................................9

6. Design Alternatives.................................................................................................................................16

7. References................................................................................................................................................17

**INTRODUCTION**

Inventory Management System is a web application used by the inventory managers

to track and monitor all the business transactions across locations. Such tracking can be manually carried out easily if there are multiple locations and can easily go through the record. Being at the location before auditing and monitoring in mandatory in each of the locations.

**PURPOSE**

This project is just a proof of concept of some concepts together in form of enterprise application. Such as:

* How applications in the enterprise integrates with each other’s.
* How to map database tables to objects.
* How authentication and authorizations work.
* How can we use AOP?
* How to handle batch processing.
* N tier model architecture.
* Make use of emailing as notification.

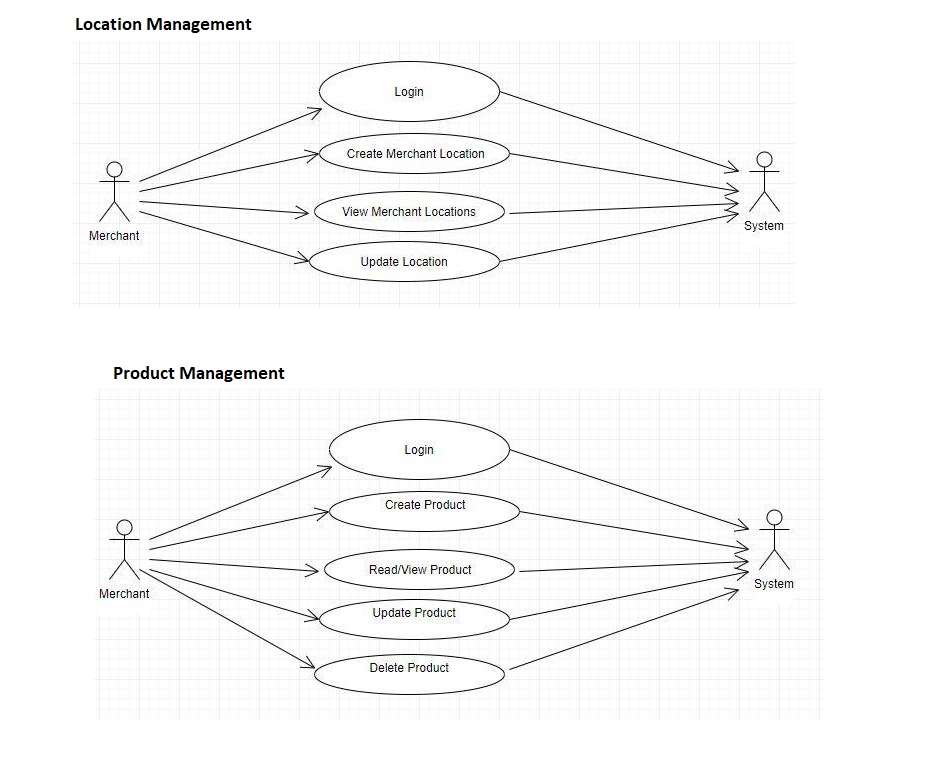
**TECHNOLOGIES USED**

To achieve the mentioned functionality, we applied technologies such as

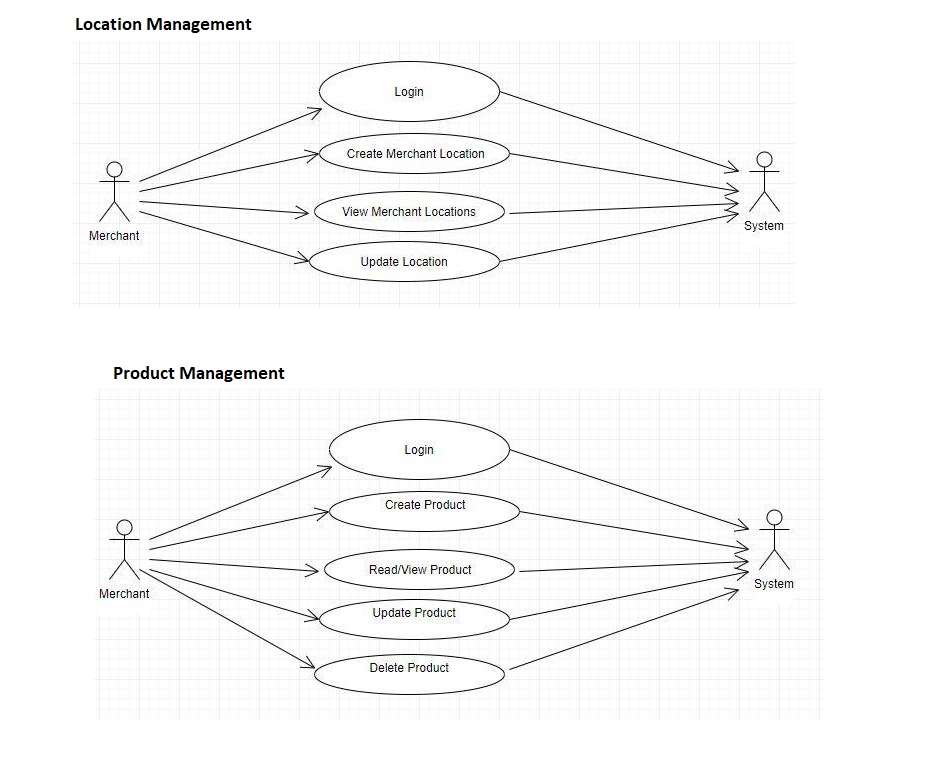
* + AOP for measuring the performance.
  + Remoting (AMQP) to send email
  + Spring security for authenticating users and differentiate between their authorities.
  + Hibernate and JPA for mapping database to objects
  + Spring Batch for creating Sales persons.
  + Email for sending confirmation to customer
  + Spring MVC for building the core system functionalities (i.e. add merchant, add salesperson, sale product)

**USE CASE DIAGRAMS**

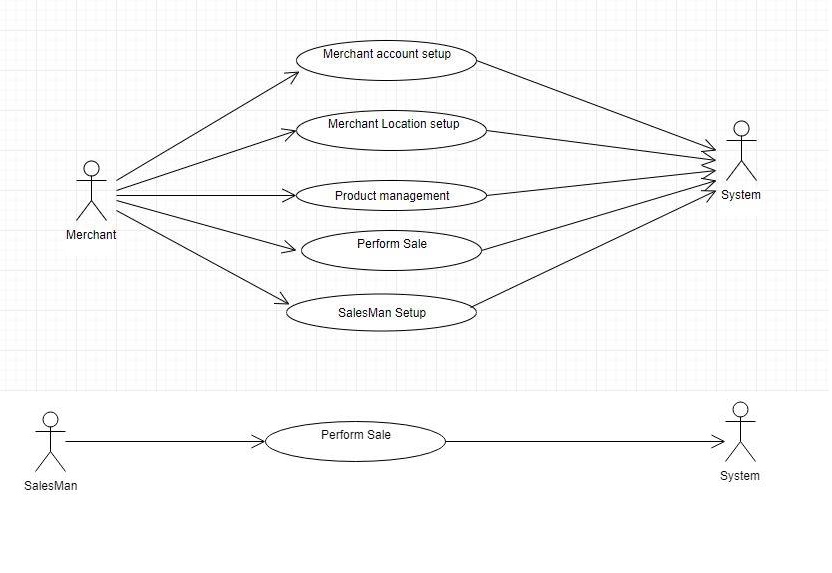
**Location Management**



**Product Management**

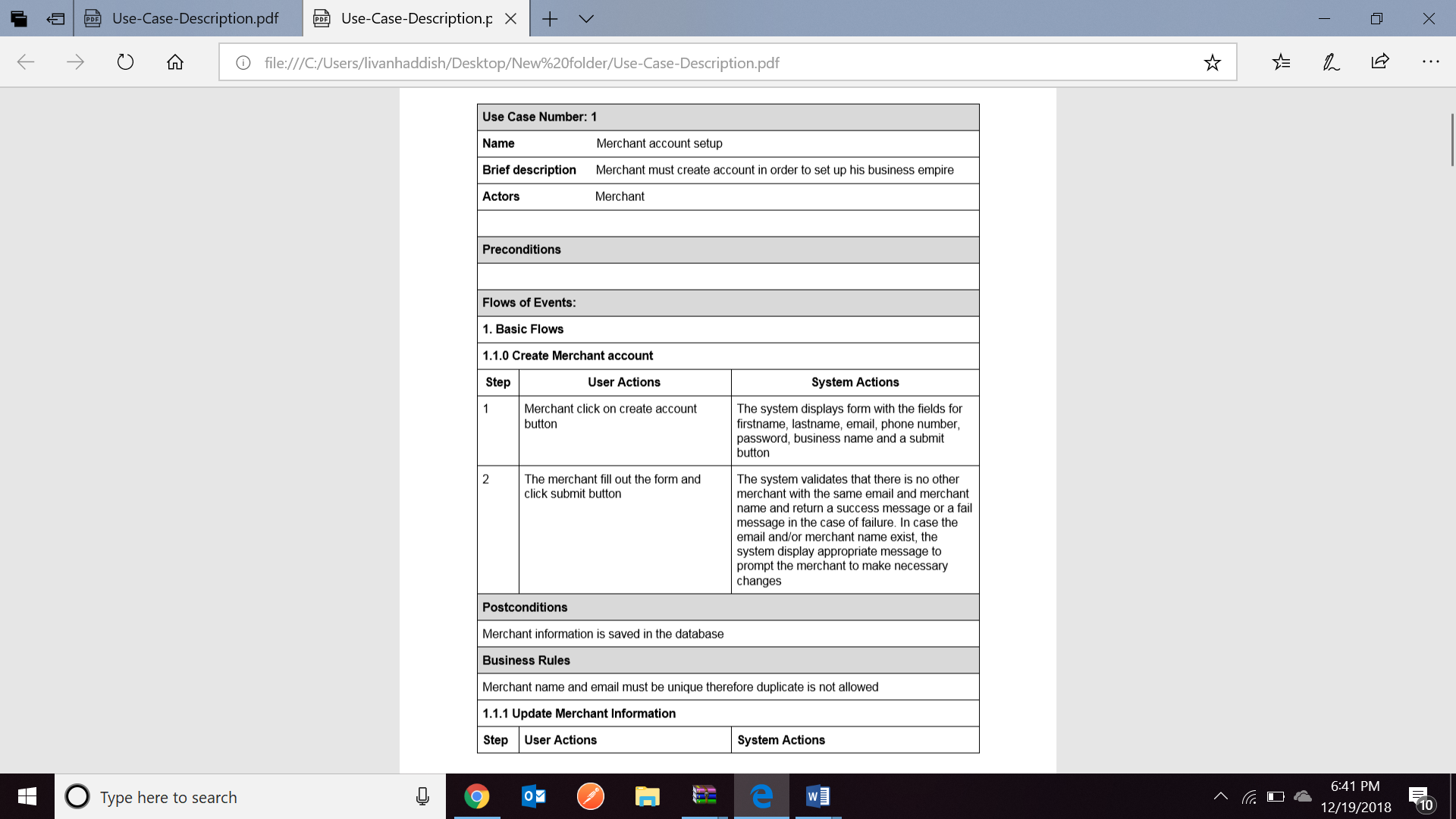


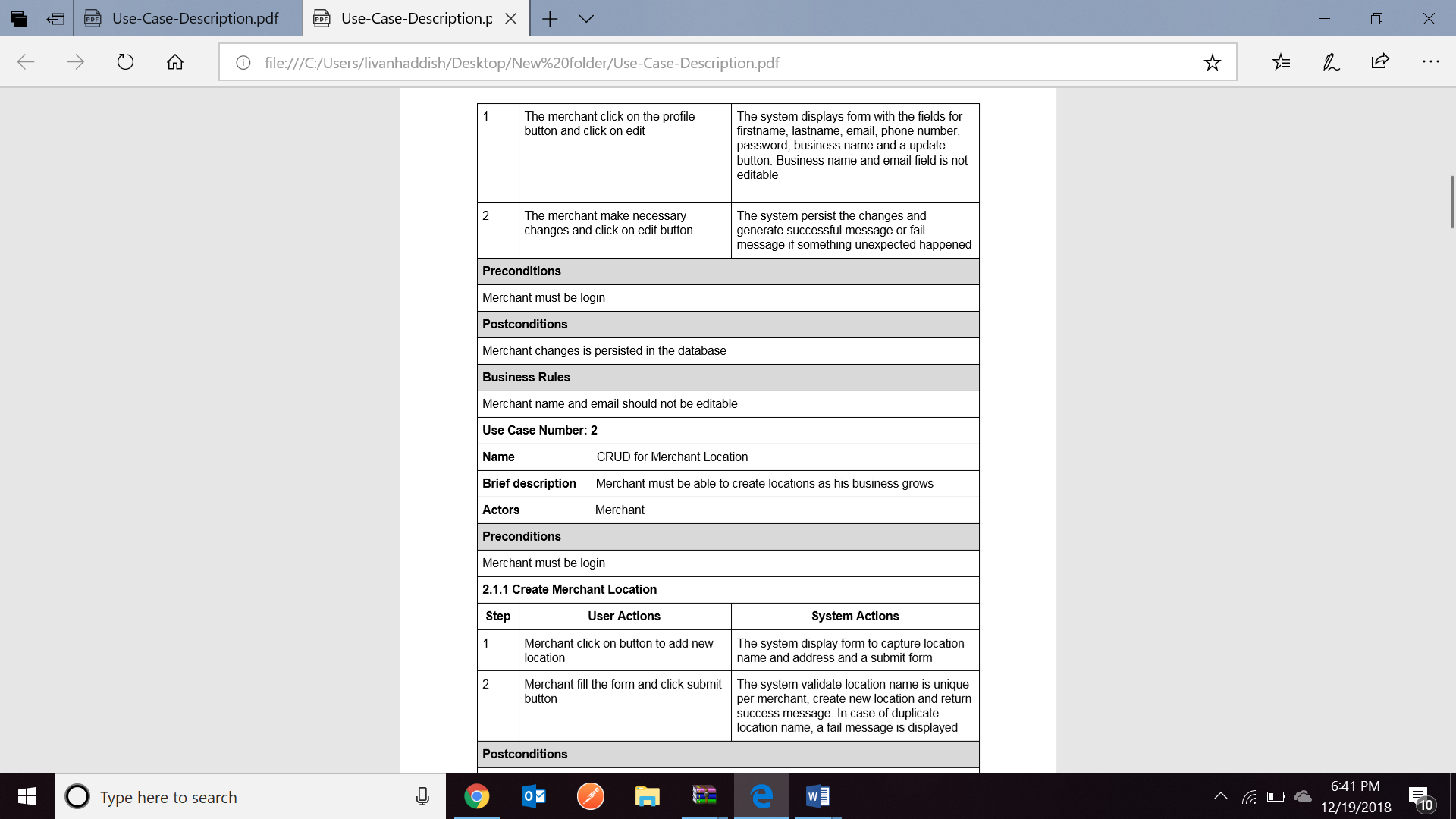
**System Management**



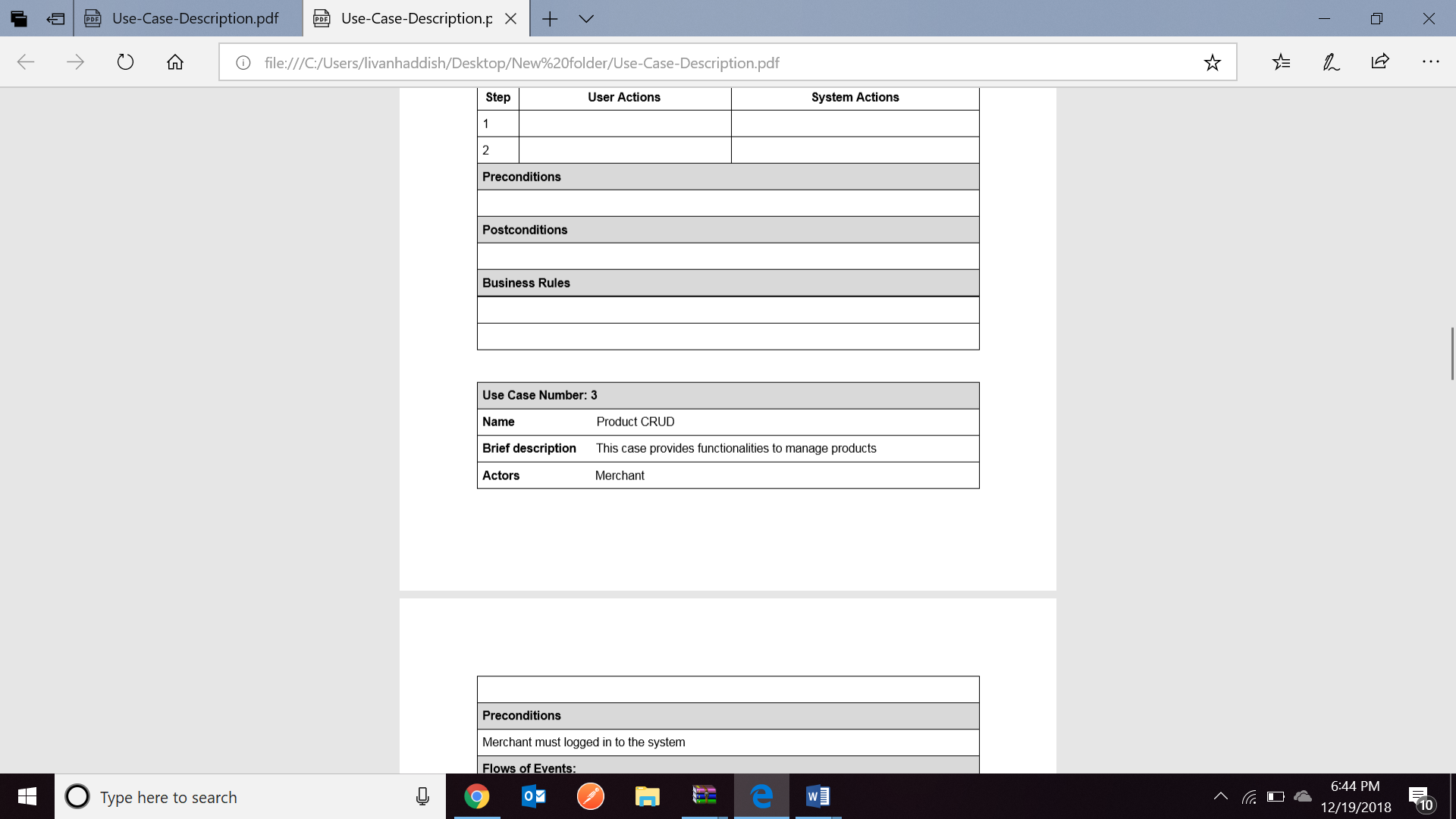
**SYSTEM USE CASES**

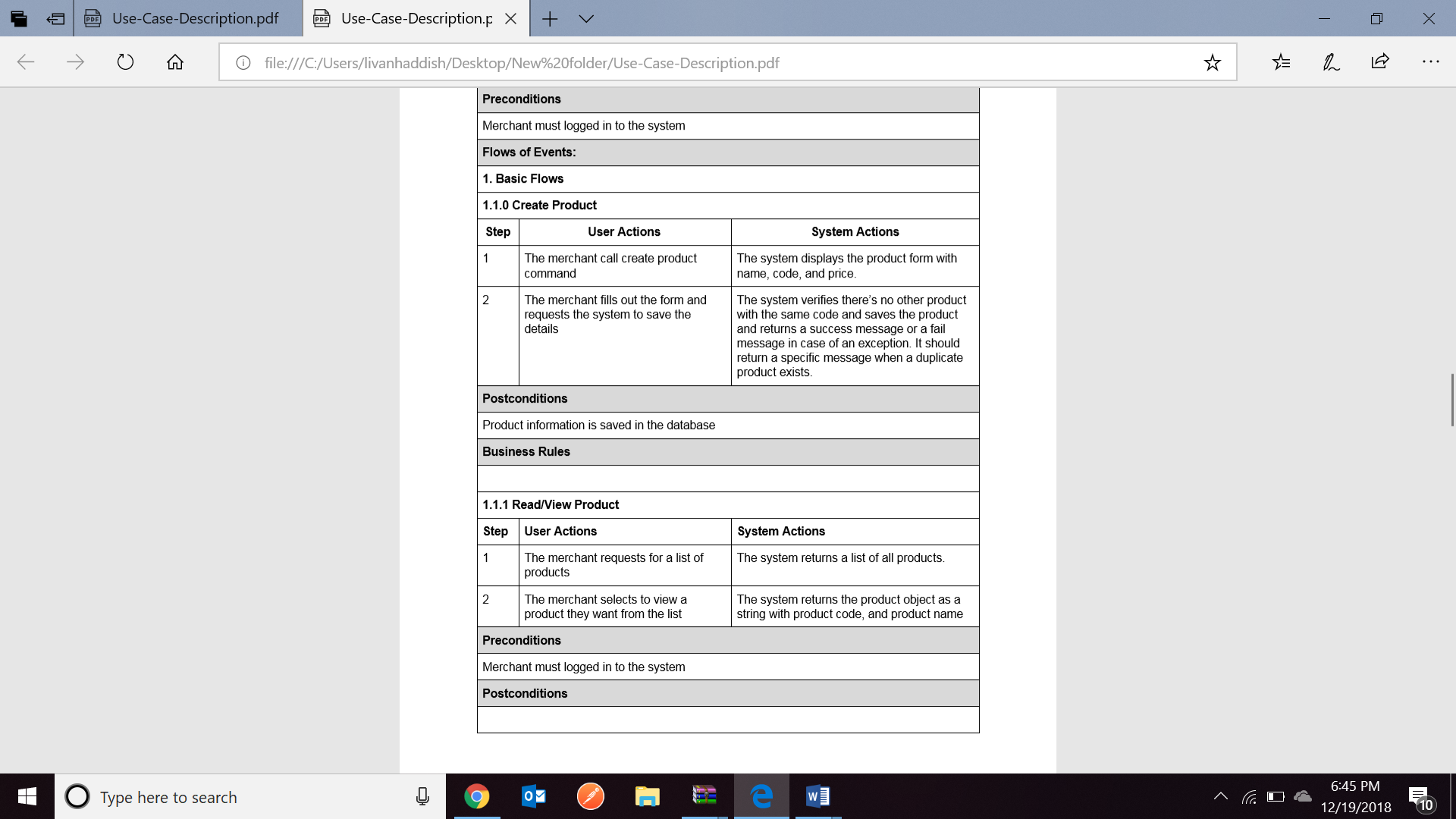
1. **Use case - Account Setup**

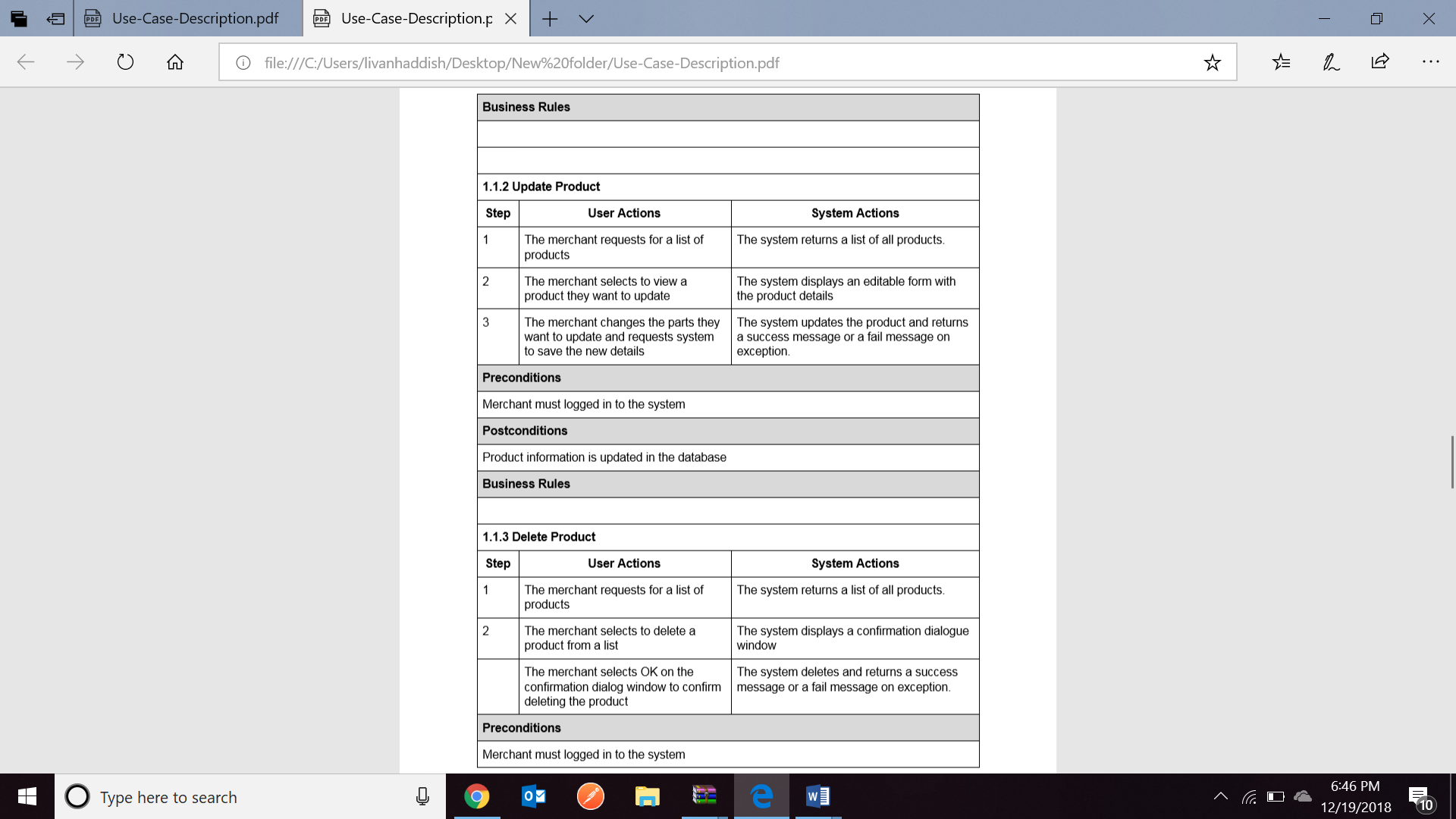


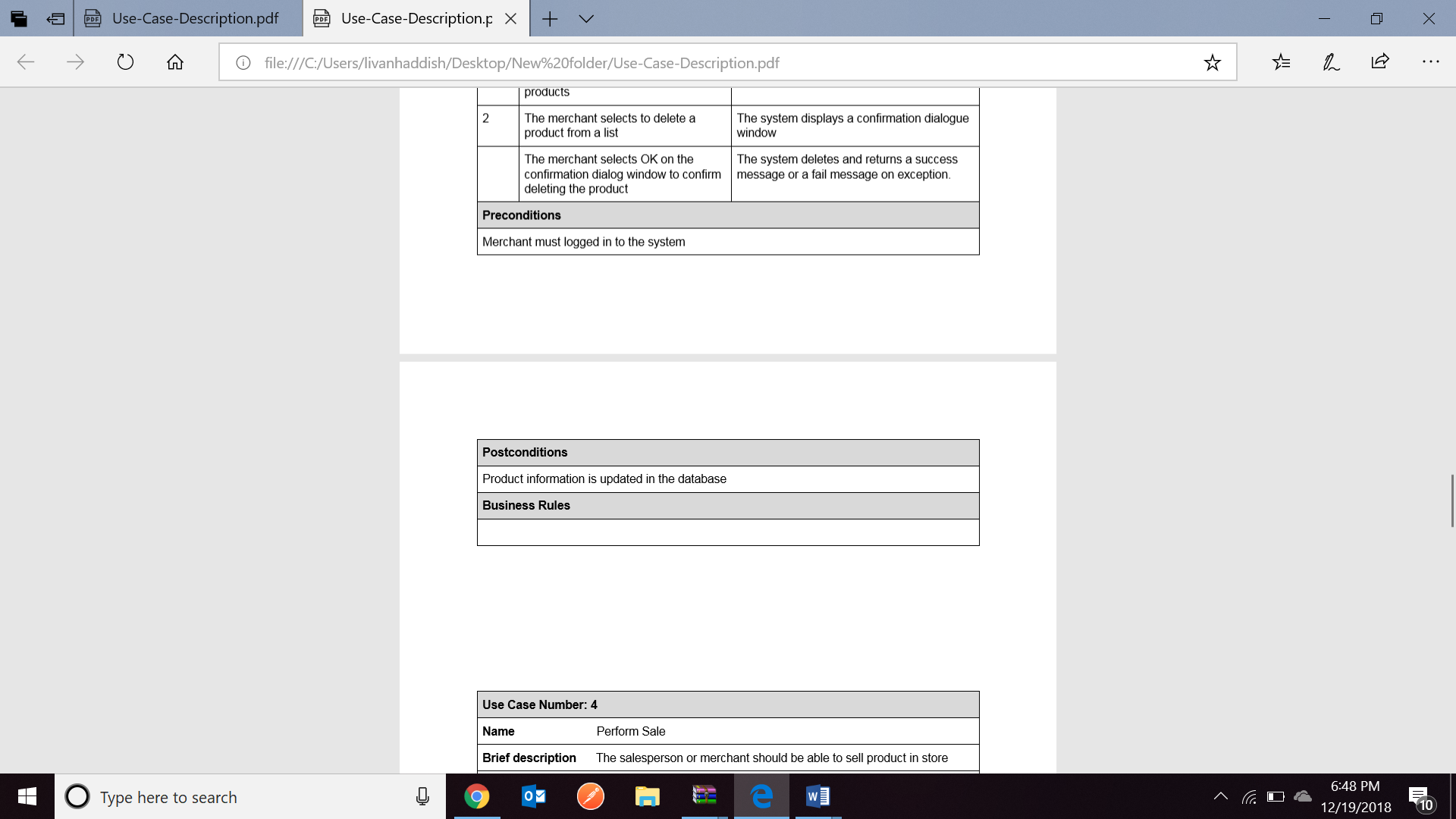


1. **Use Case – Product CRUD**

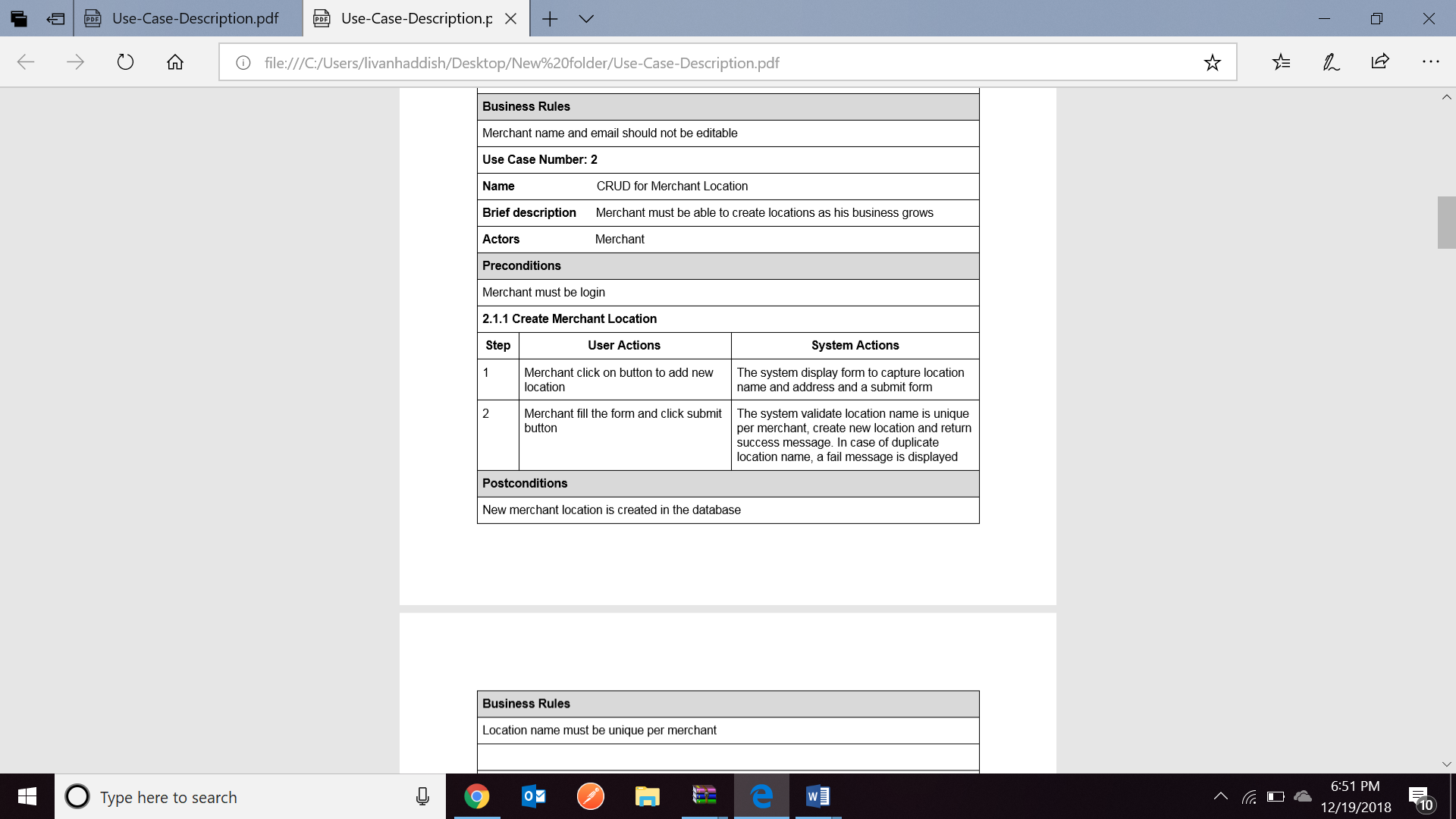


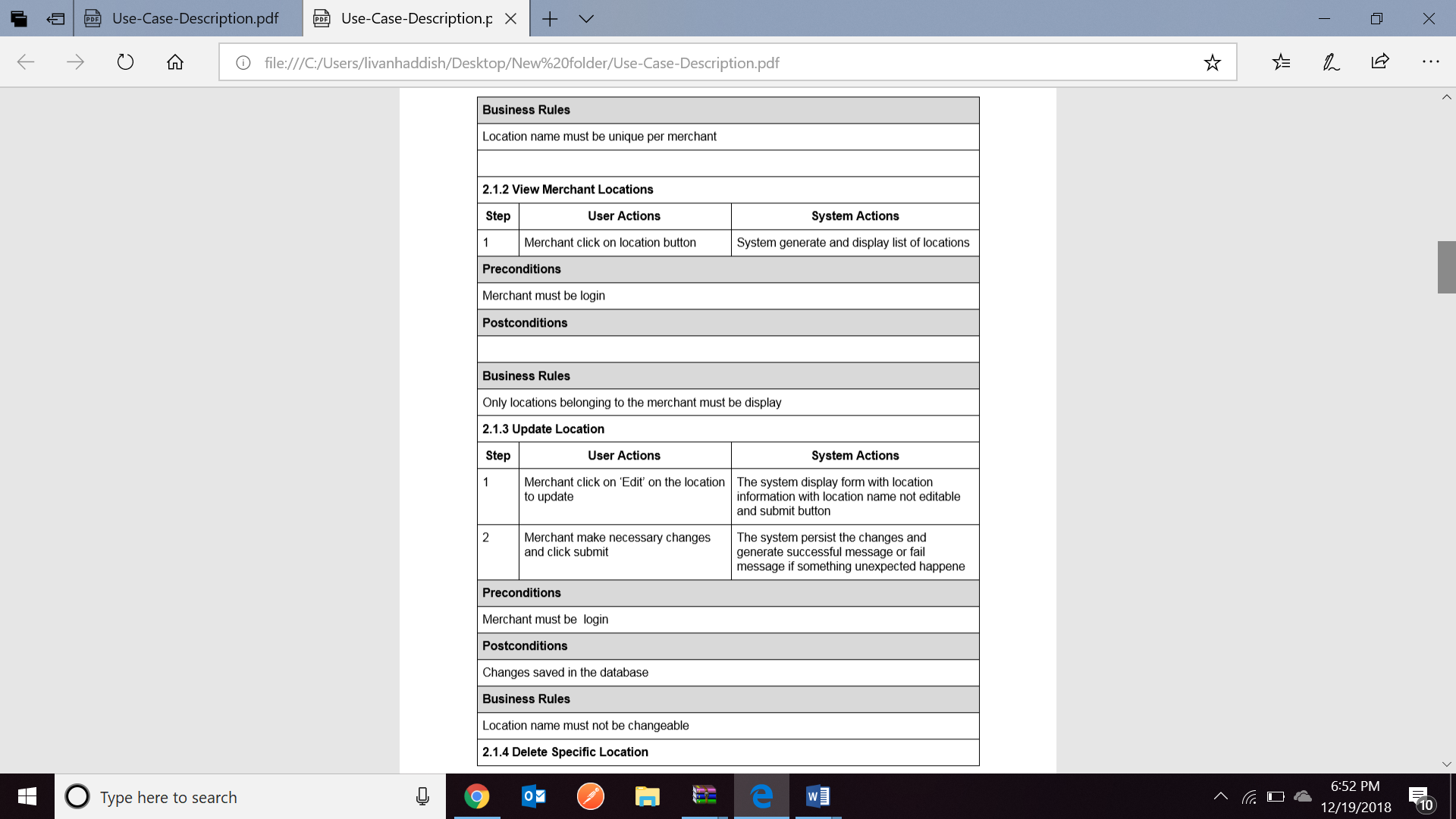


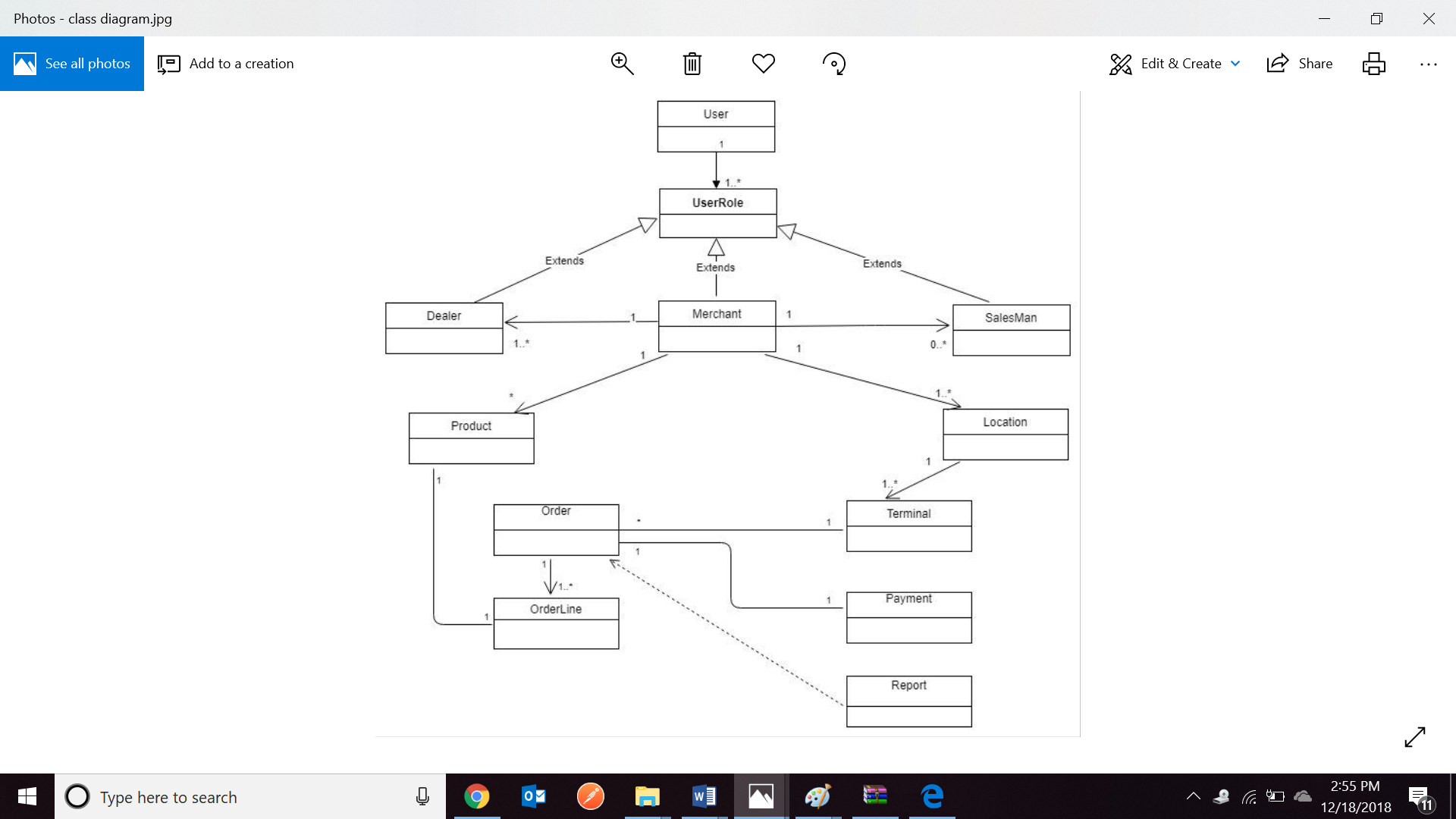


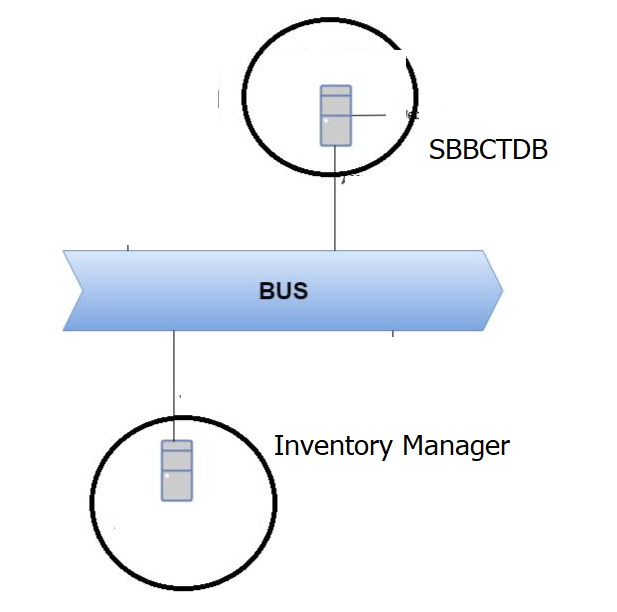


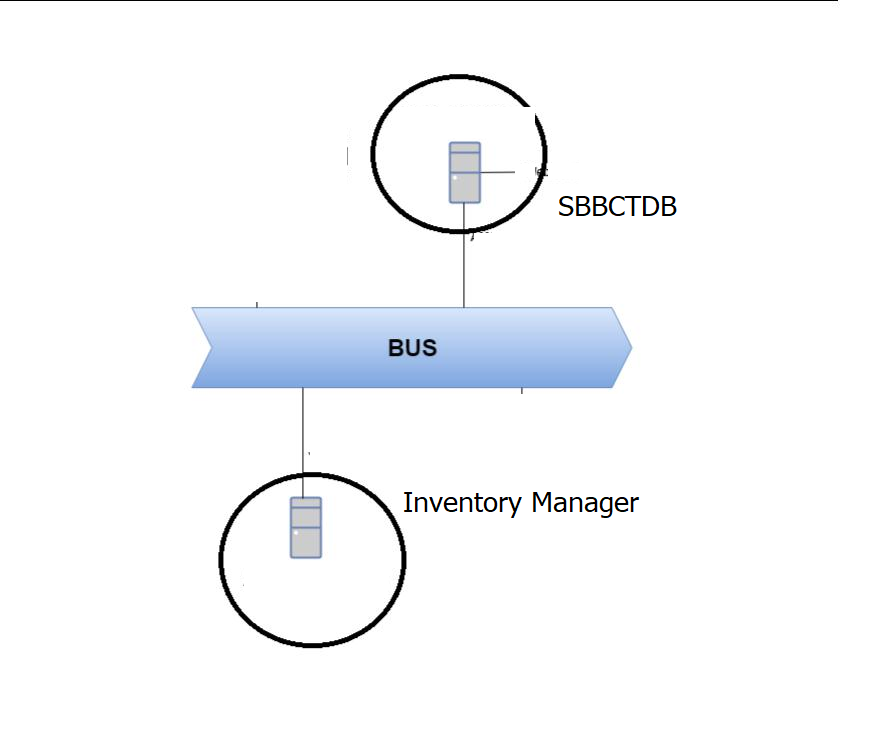
1. **Use Case – Merchant Location CRUD**



**CLASS DIAGRAM**



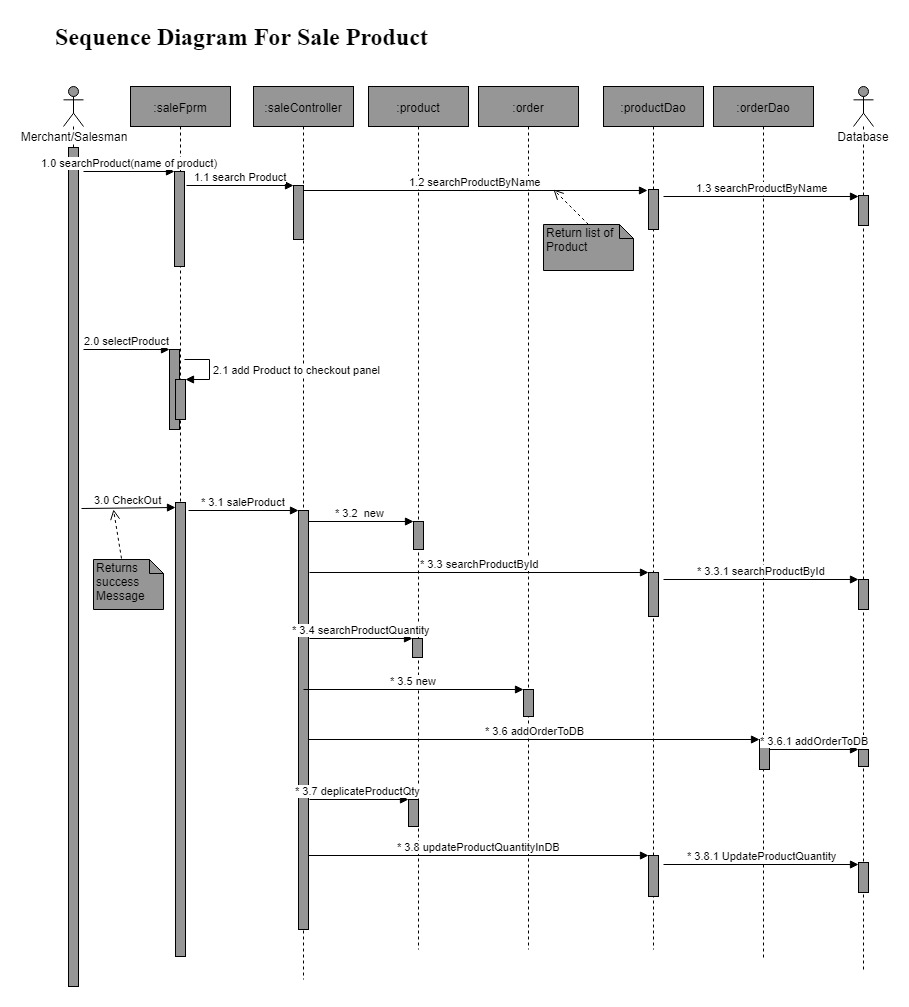


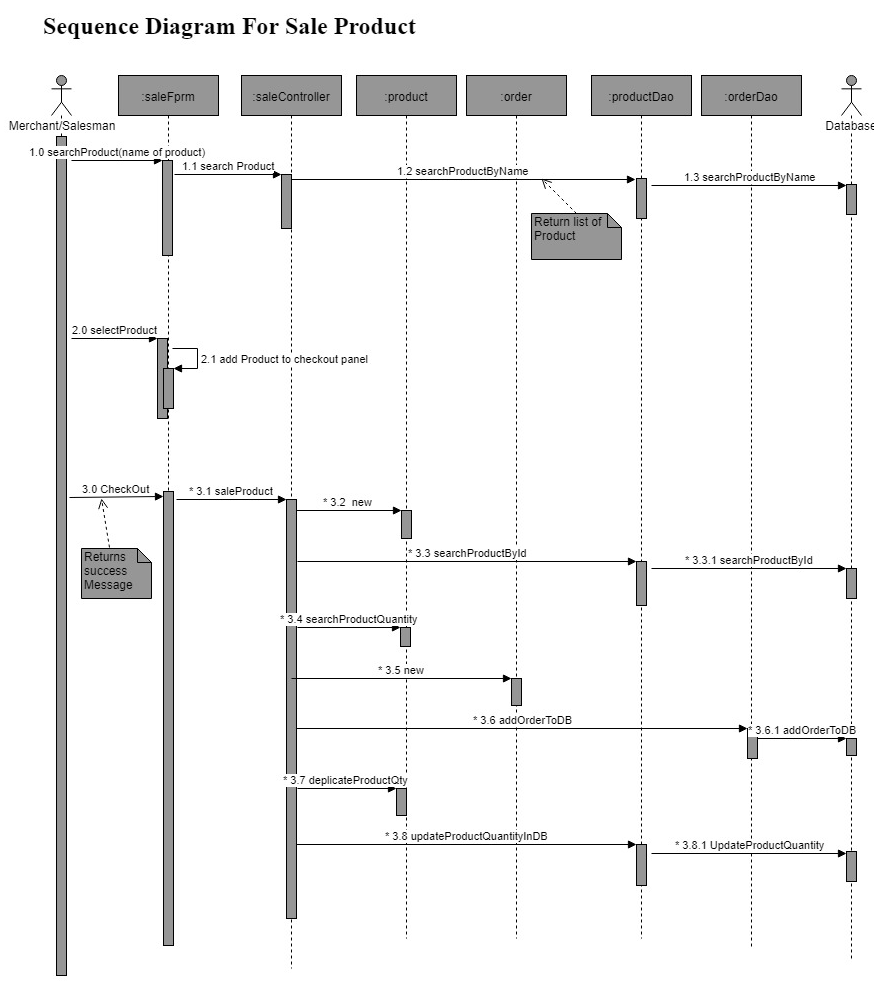
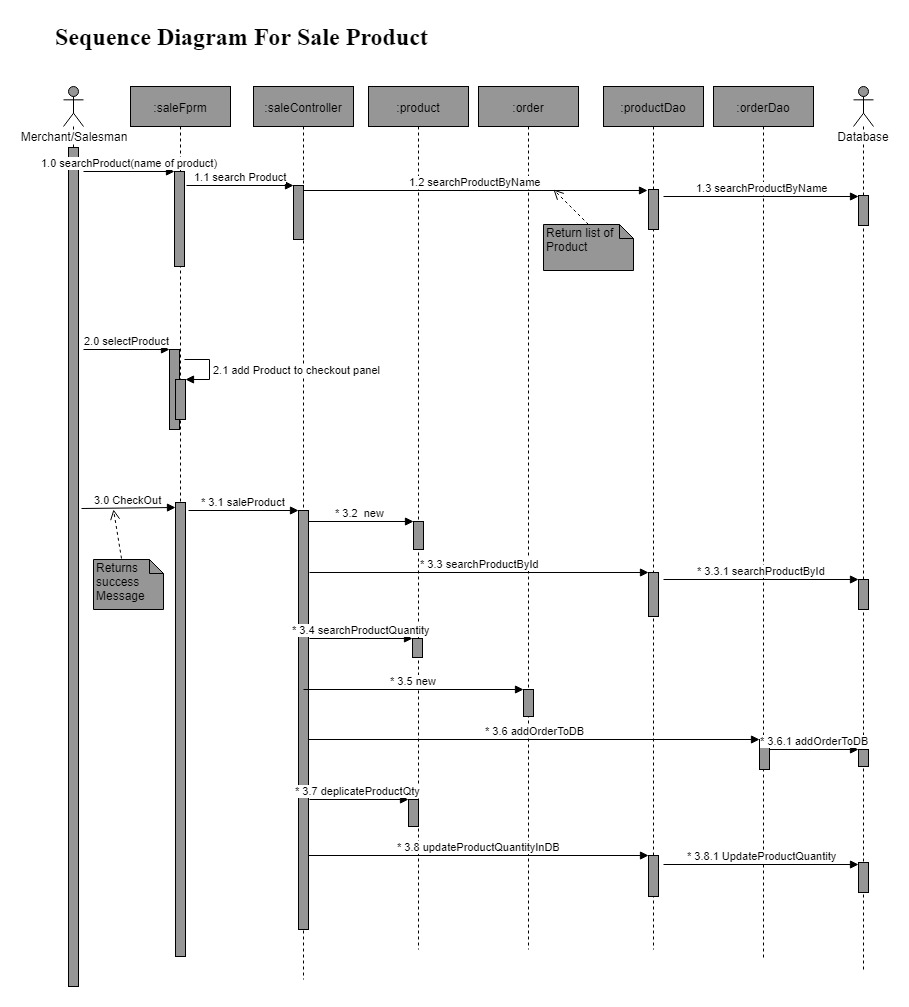


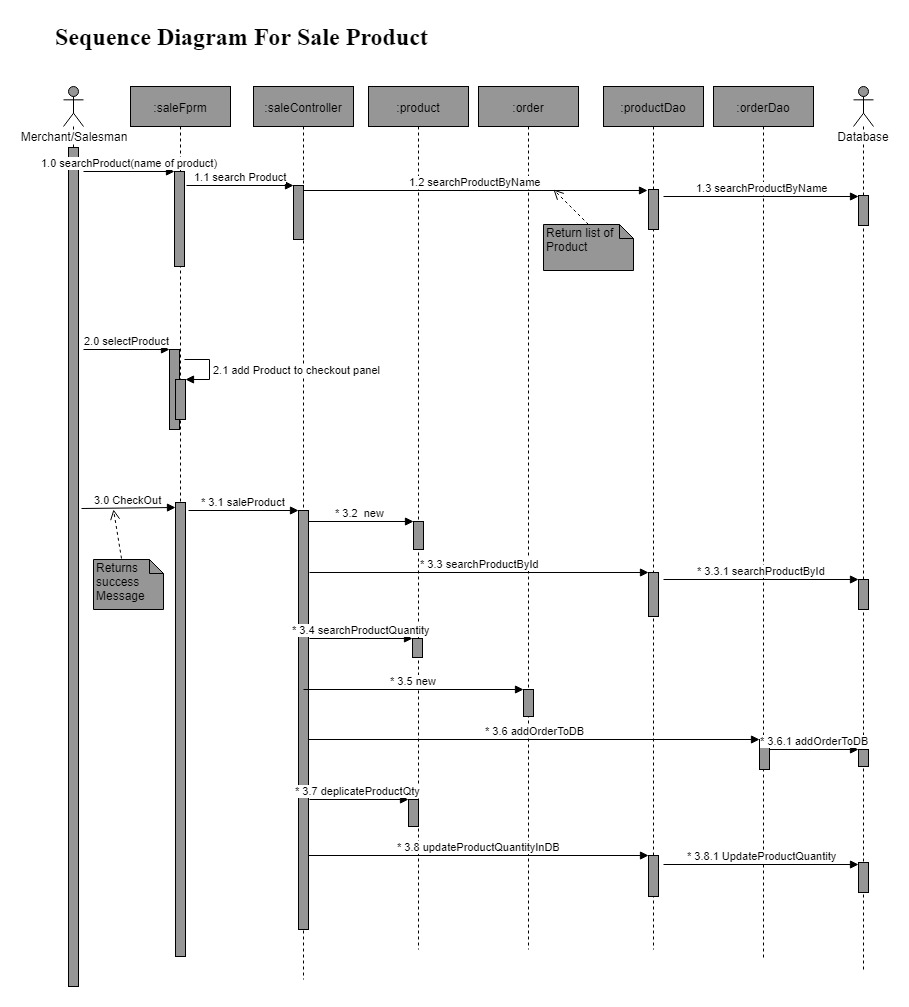
**RESPONSIBILITIES/ROLES:-**

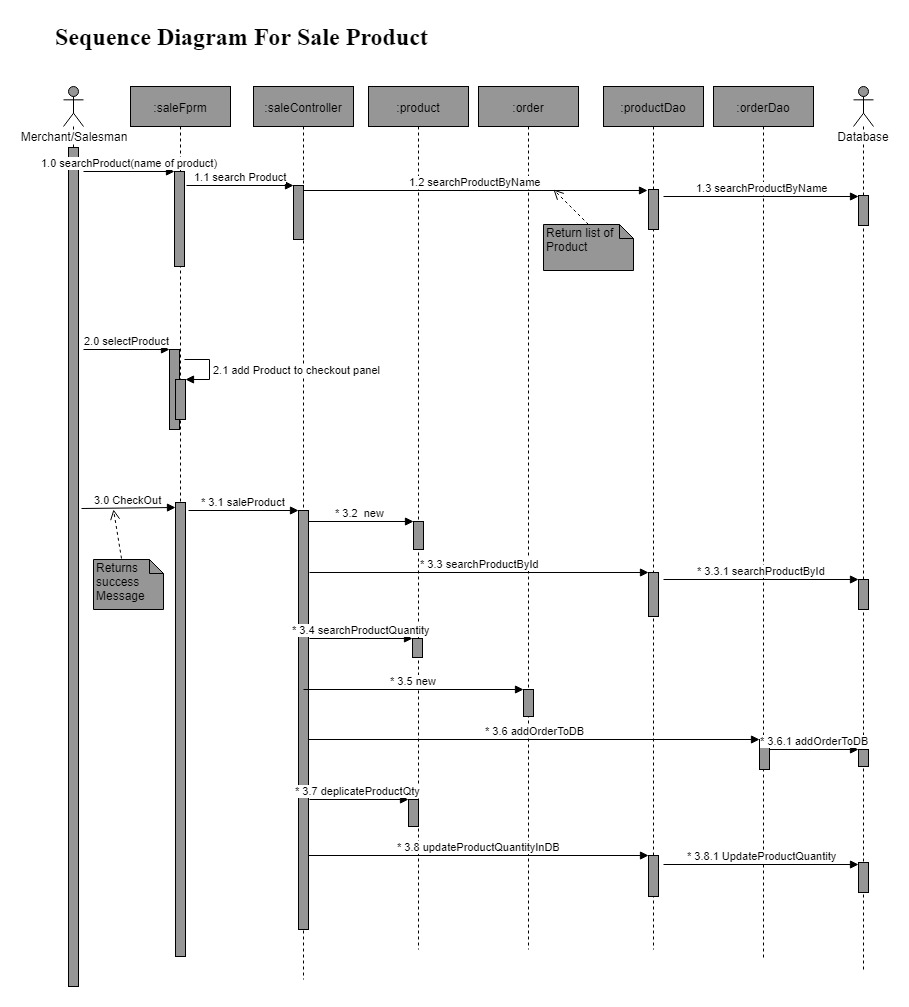
|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Responsibilities** |
| Merchant | Merchant create, update products, , locations,dealers and salesperson | Merchant is responsible for creating and updating products, locations, dealers and salesperson. |
| Dealers | Dealer supply the merchant with products in different locations and upload report in the solution for merchant perusal |  |
| Salesperson | Salesperson login to the system and perform sale based on customer selection |  |

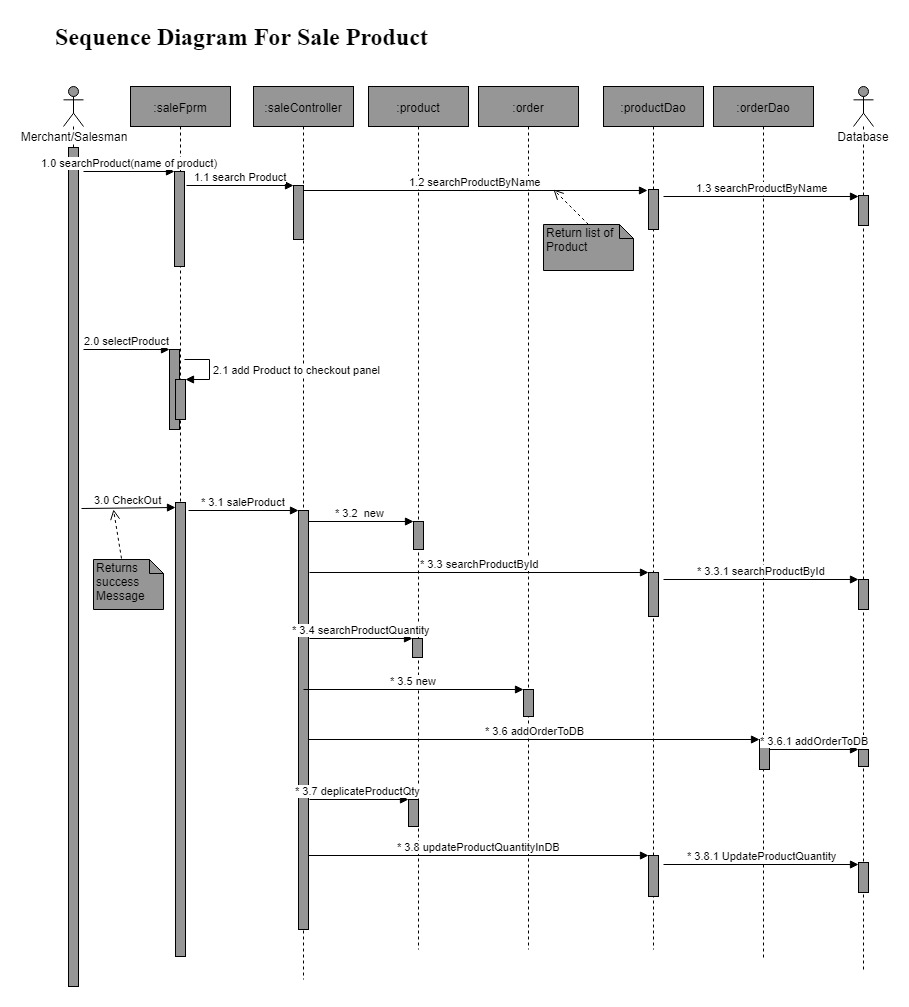
**Interaction Diagram**

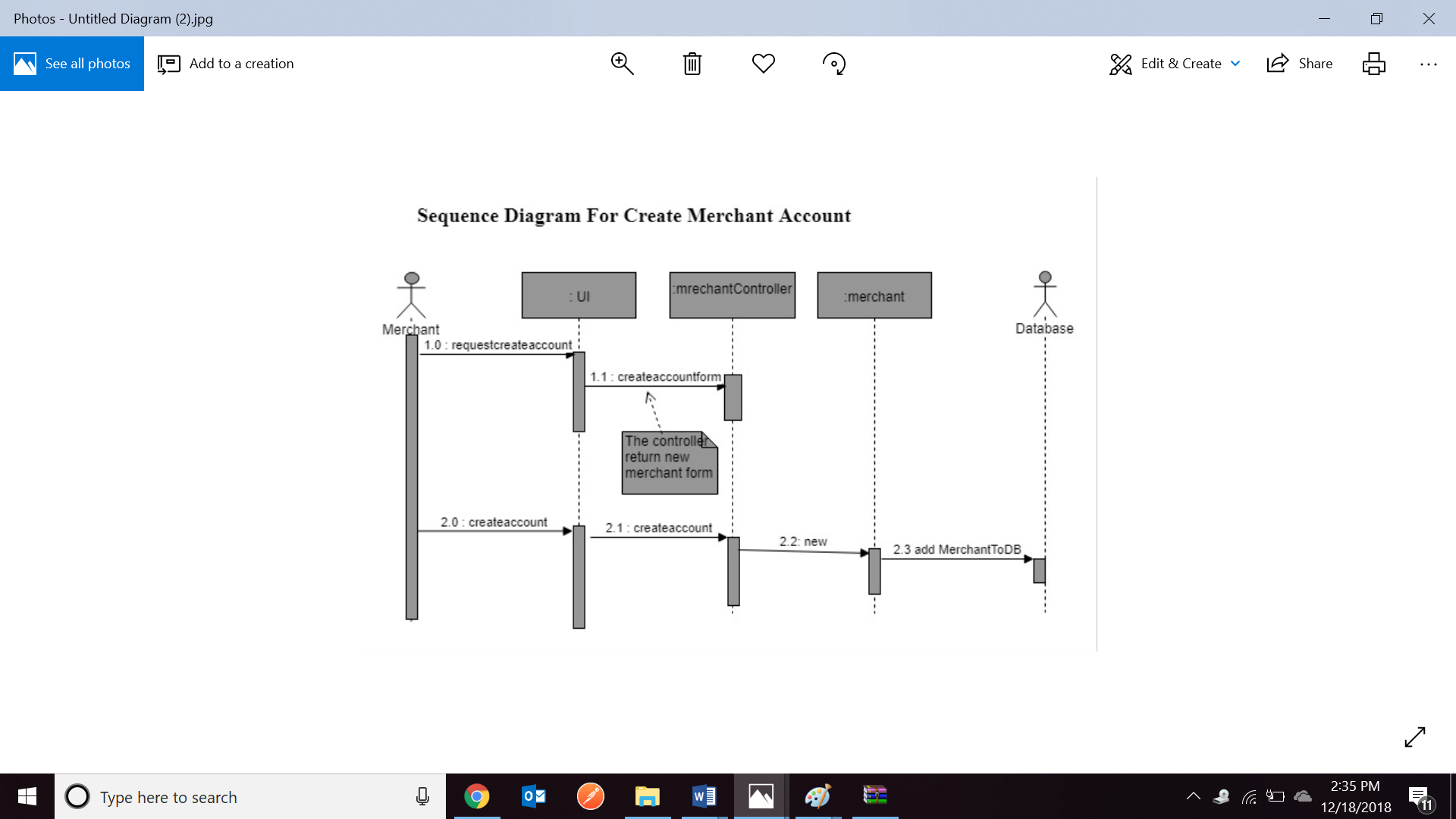
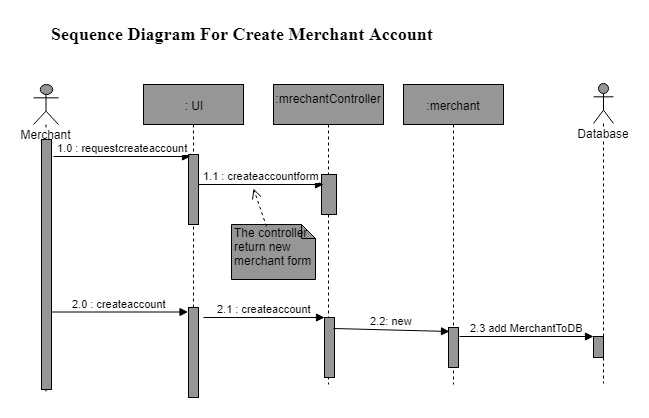


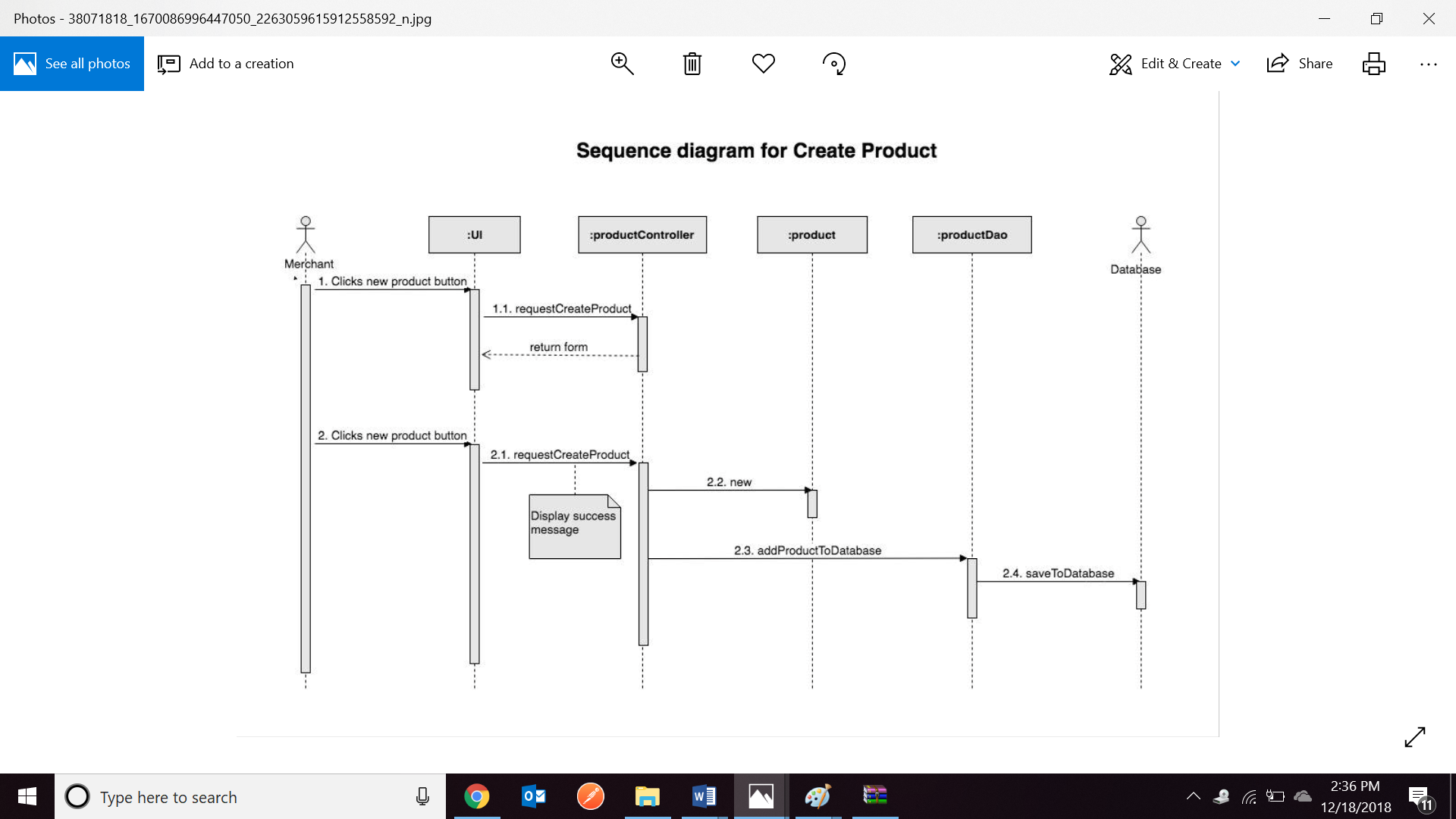


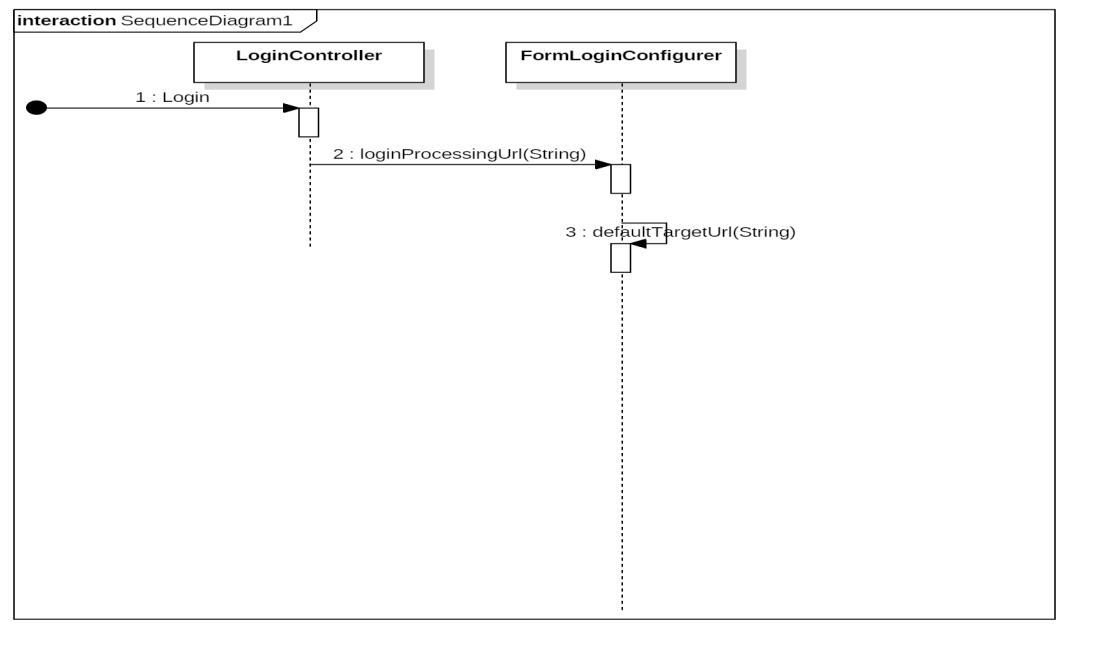


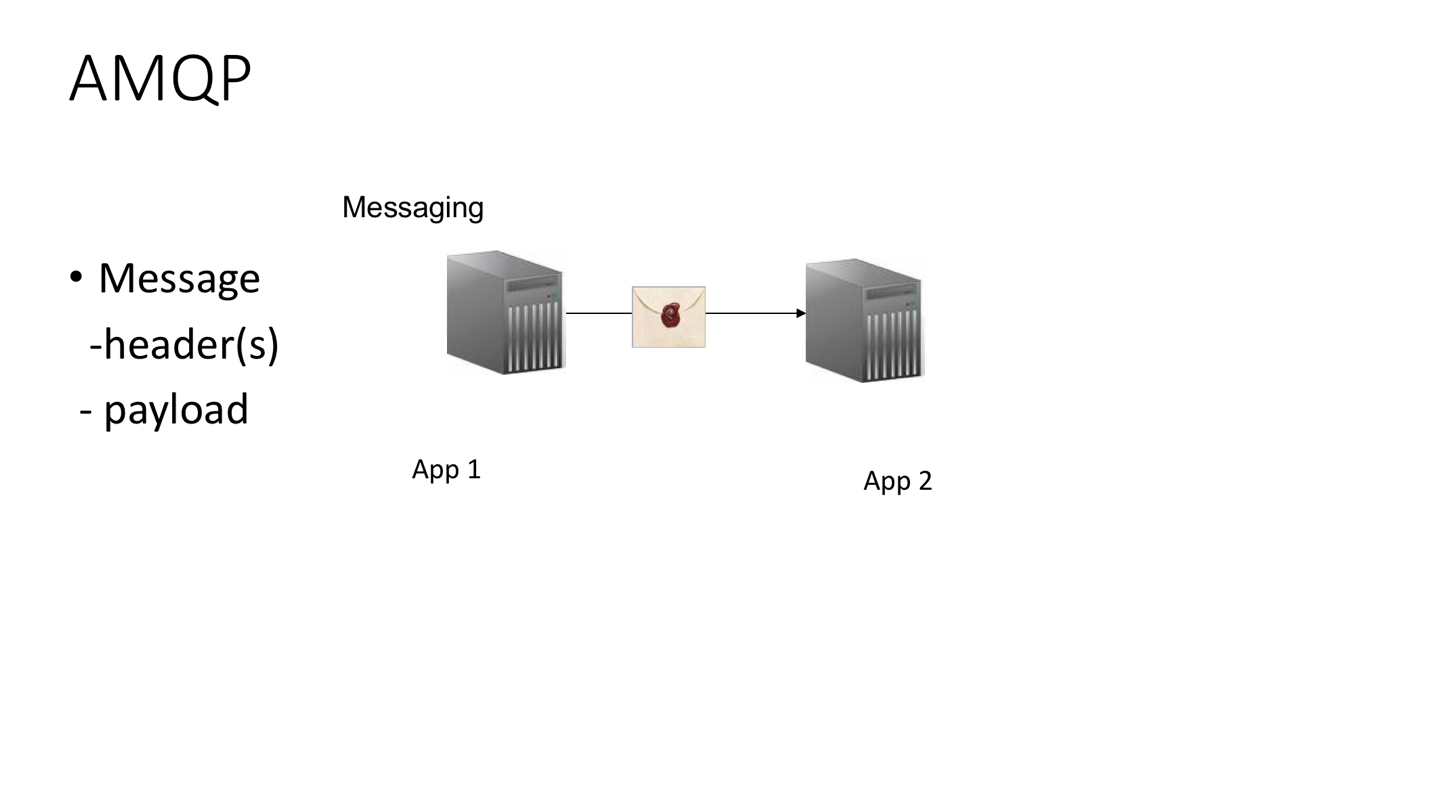






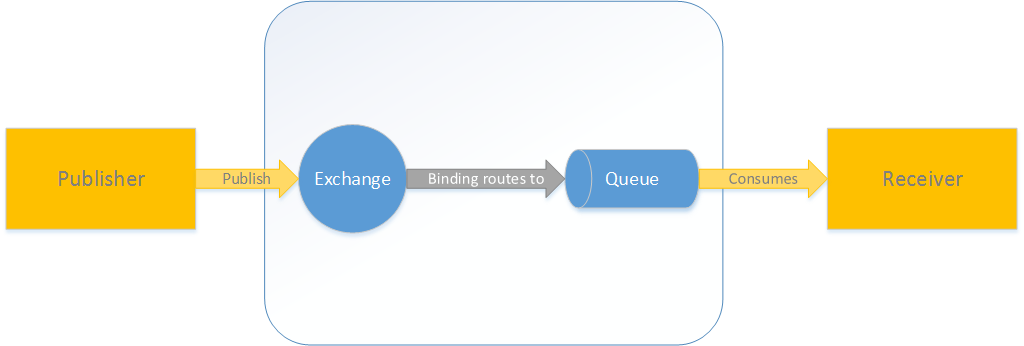




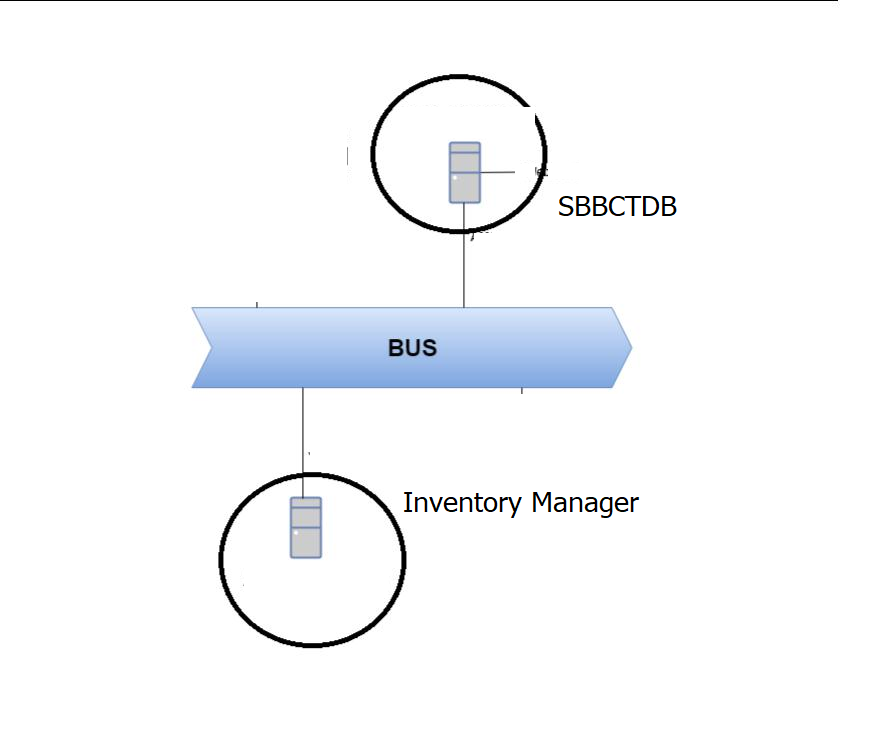


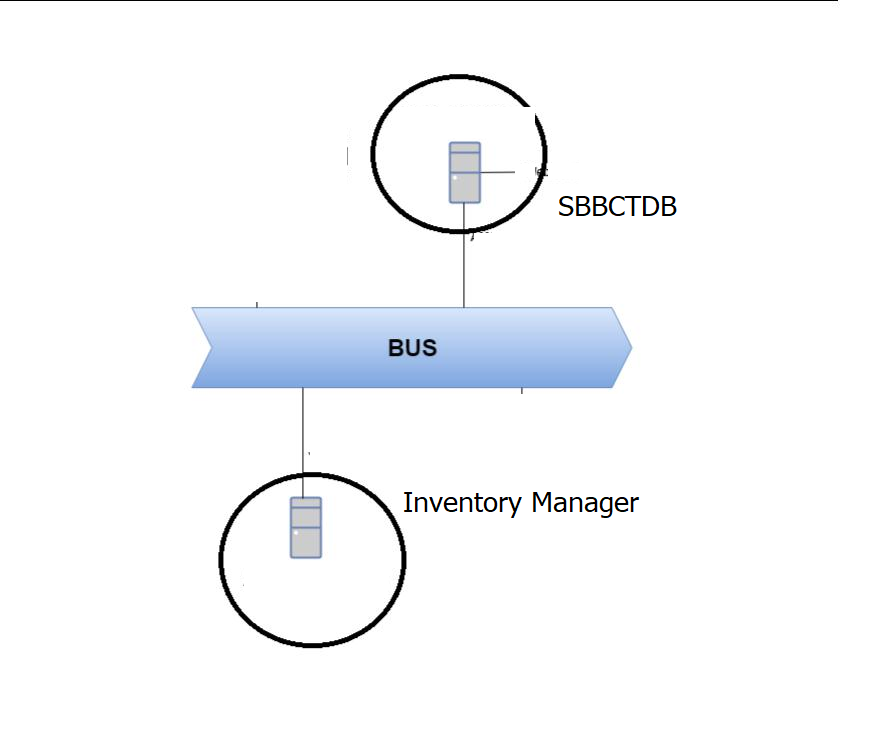
AMQP

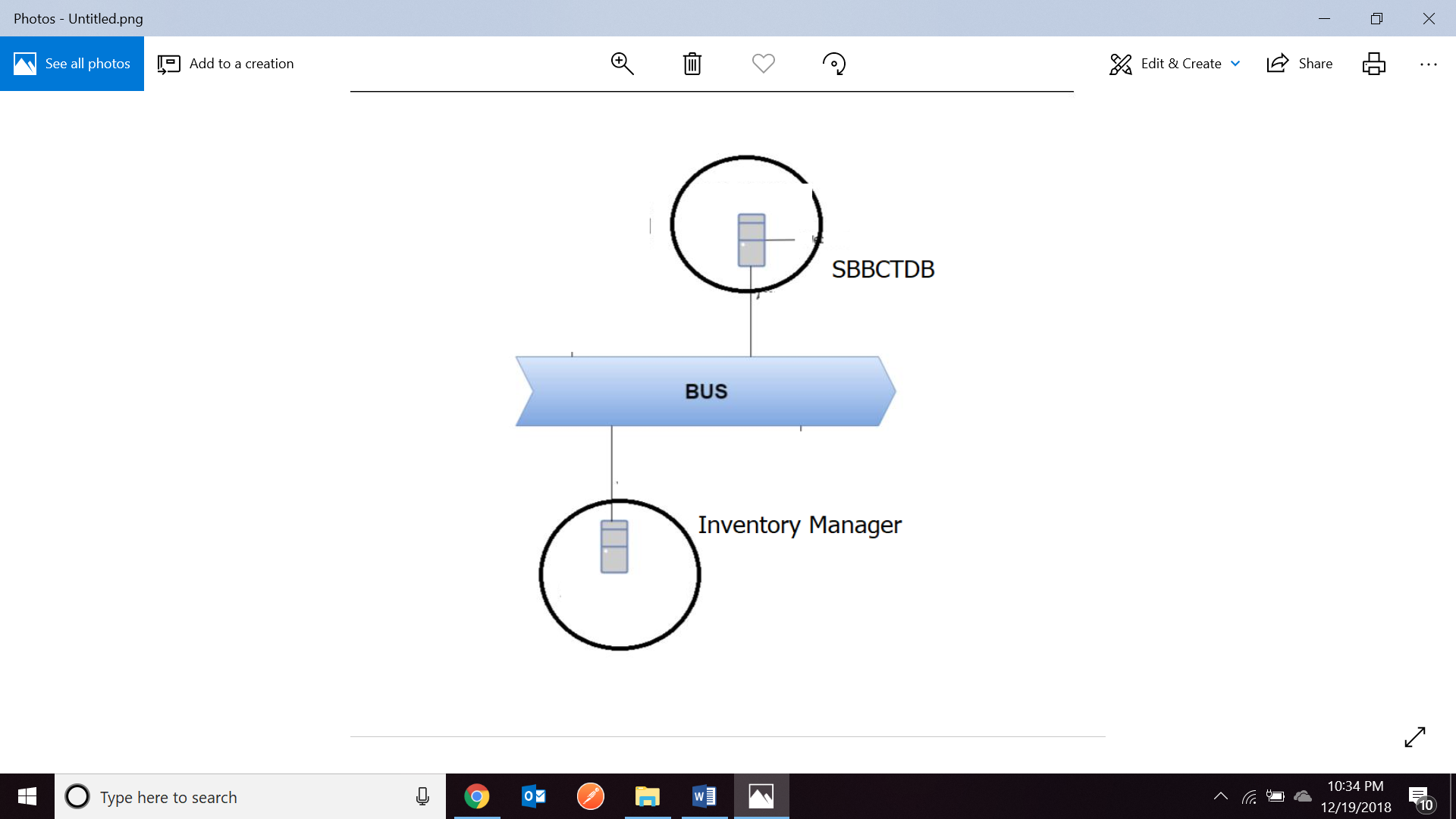
**Messaging**



**AMQP MESSAGING**







**USE CASE**

* New Buyer is added to ProductChain(App 1)

- Buyer details need to be sent to our Second application(App 2)

- App1 (producer)

- App 2(consumer)

- both use RabbitMQ

* Queue
* Topic Exchange
* RabbitTemplate

**AMQP Concepts**

• Exchanges - Message routing agents

• Bindings - binds/maps a queue & exchange

• Routing Key - optional attribute to customize binding/routing

• Queues - Message placeholders

**ISSUES:**

**-**while we were working with the AMQP messaging when we send an object we can see it in the rabbitmq (queue) but the message wasn’t displayed in our second application (consumer) and we tried it by sending a plain string and it was working and was consumed. The reason was the package name was different on the producer and consumer classes.

**References**

https://www.udemy.com/learn-rabbitmq-asynchronous-messaging-with-java-and-spring/learn/v4/t/lecture/9608714?start=255

Spring Batch

<https://docs.spring.io/spring-batch/trunk/reference/html/>

Spring Security

<https://docs.spring.io/spring-security/site/docs/5.0.3.RELEASE/reference/htmlsingle/>

Spring AOP

<https://docs.spring.io/spring/docs/2.5.x/reference/aop.html>

<http://www.baeldung.com/spring-aop>

Spring Security

<https://www.udemy.com/spring-hibernate-tutorial/learn/v4/t/lecture/7413896?start=135>

class demo