

**โครงงาน**

KAB Phone

**จัดทำโดย**

6604062636348 นายธีรภัทร โคตะนารถ

6604062636453 นายพงศ์ศิริ เลิศพงษ์ไทย

6604062616215 นายอิทธิวรรธน์ ขุนพิชัย

**เสนอ**

ผู้ช่วยศาสตราจารย์ ดร. คันธารัตน์ อเนกบุณย์

**วิชา System Analysis and Design**

ภาคเรียนที่ 1 ปีการศึกษา 2567

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

**สารบัญ**

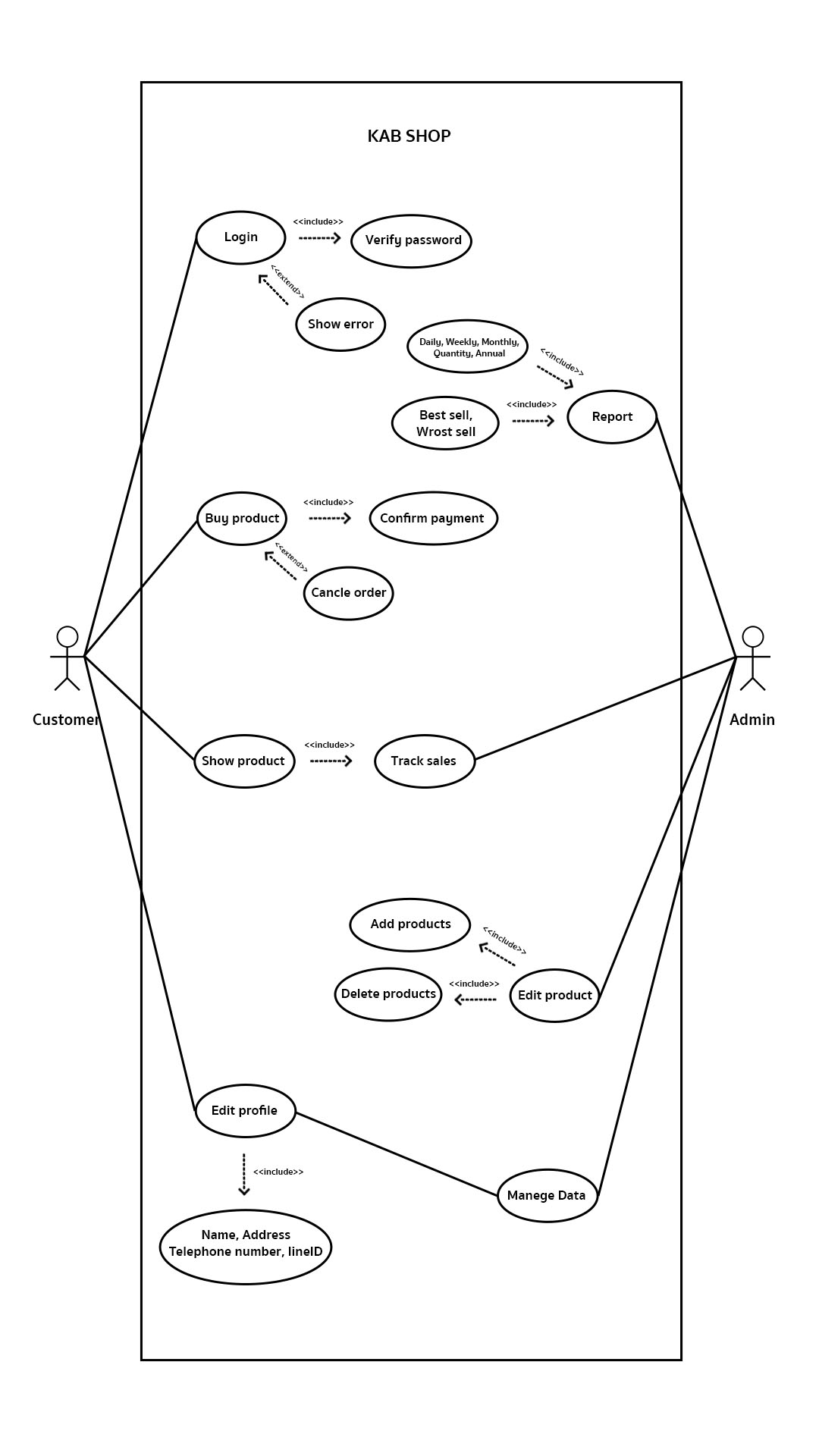
**หน้า**

1. Requirement 1
2. Use case 2
3. 2 Fully use case description 3 - 4
4. Context diagram 5
5. DFD Level 0 6
6. 2 Sequence diagram 7
7. Interface 8
8. Testing 9

**Requirement**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Stakeholder** | **Hardware** | **Input** | **Output** | **Functional requirement** | **Nonfunctional requirement** | **When** |
| Admin |  | New phone data |  | An admin adds new phone to stock or remove outdated phone from stock |  | They launched a new phone or there is an outdated phone |
| Admin |  | Tracking number |  |  |  | Paid order occurred |
| Customer | * Payment * Address * Tel no. |  | * Updated stock * Invoice receipt |  | When customer add item to basket. It will not deduct from stock until payment success | Daily |
| Customer | * Name * Address * Tel no. * Line ID |  |  |  | Customer can edit name, address, Tel no., or Line ID | Daily |

**Use case**

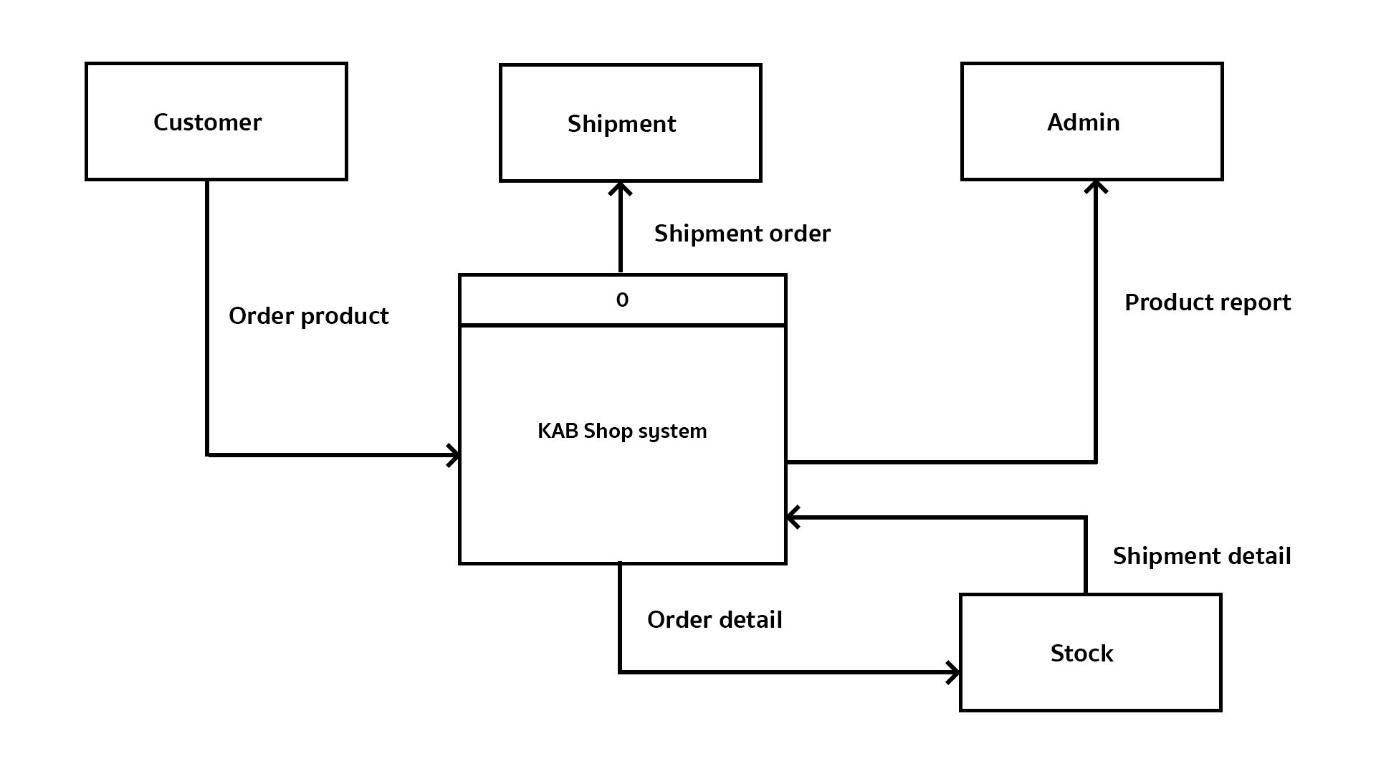
****

**2 Fully use case description**

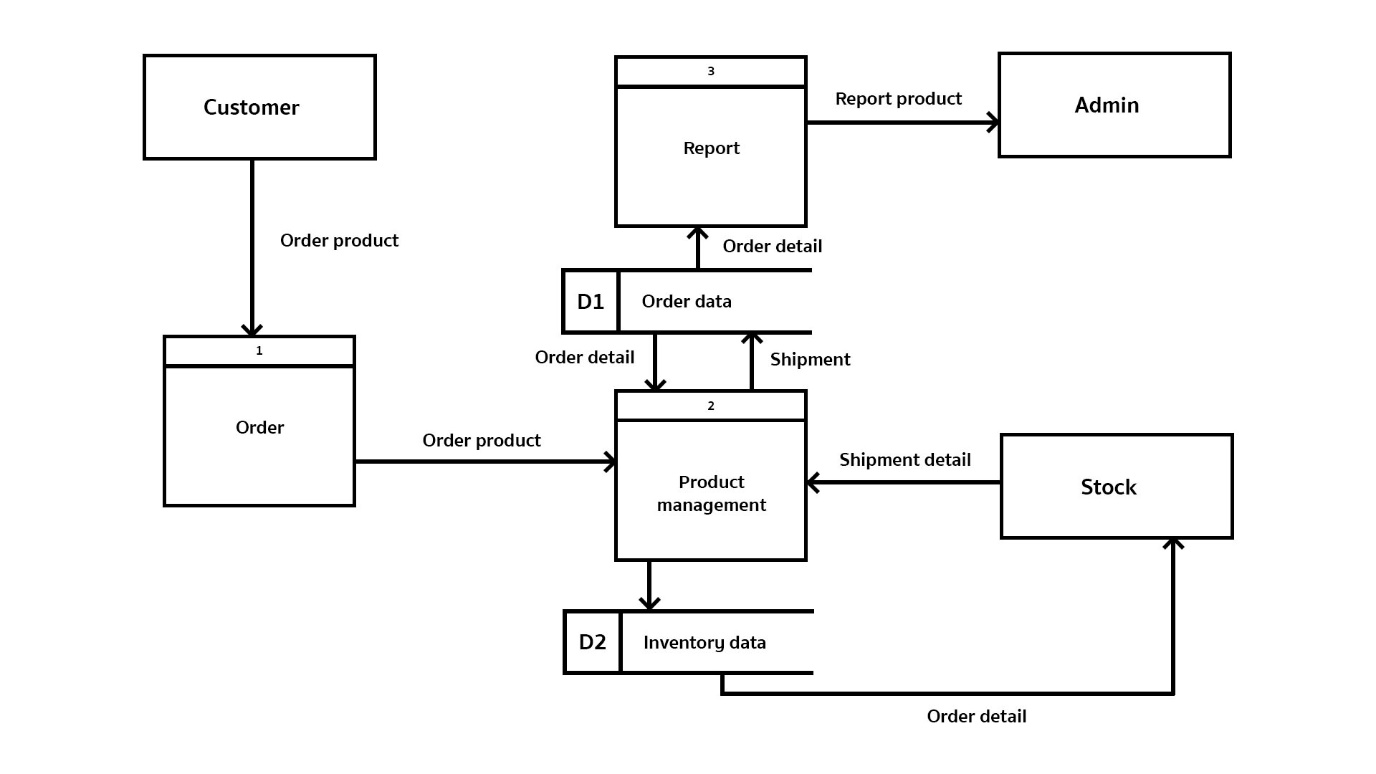
|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Add item to basket | |
| **Scenario:** | Customer adding item to basket and see the total price. | |
| **Triggering Event:** | Customer adds an item into the basket. | |
| **Brief Description:** | First, customer click add an item into the basket. Then the system will return item price and item description. Then calculate total price. | |
| **Actors:** | Customer | |
| **Related Use Cases:** | Including: Stock Check, Item Description | |
| **Stakeholders:** | Admin: to provide description related to that item from stock. | |
| **Preconditions:** | Item that customer added. | |
| **Postconditions:** | * Show an error if item is out of stock. | |
| **Flow of Activities:** | **Actor** | **System** |
|  | 1. Customer is looking for a wanted item from the store front page. 2. Customer clicked add item to the basket.   2a. if out of stock, go back to step 1.   1. Customer adds quantity they want into the specified item in the basket. 2. Customer place an order. | 2.1 System check if item is not empty.   * 1. Then return the price to customer. |
| **Exception Conditions:** | 2. When item is out of stock.  3. When item is not enough in stock. | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Place Order | |
| **Scenario:** | Customer is adding item into basket, then make a payment. | |
| **Triggering Event:** | Add an item into the basket | |
| **Brief Description:** | At first, customer is select the prefer item. Then add it into basket. System will check if item is ready for sell. If yes user will make payment, and done. | |
| **Actors:** | Customer | |
| **Related Use Cases:** | Including: Stock Check, Item Description, Payment | |
| **Stakeholders:** | Admin: to provide stock, and item description.  Bank: to provide payment process. | |
| **Preconditions:** | Item that customer added. | |
| **Postconditions:** | * Show an error if item is out of stock. * Show an error if payment process failed. * Show success page if everything goes correctly. | |
| **Flow of Activities:** | **Actor** | **System** |
|  | 1. Customer selects an item. 2. Customer add it into basket.   2a. if empty, goes to 1.   1. Customer confirm an order. 2. Customer making a payment.   4a. if payment failed, goes to 2.   1. Order placement success. | 2.1 System check for stock.  2.2 If stock is empty then return to customer.  4.1 System contact the payment provider.  4.2 System return payment status. |
| **Exception Conditions:** | 2. When item is out of stock.  4. When payment is not success. | |

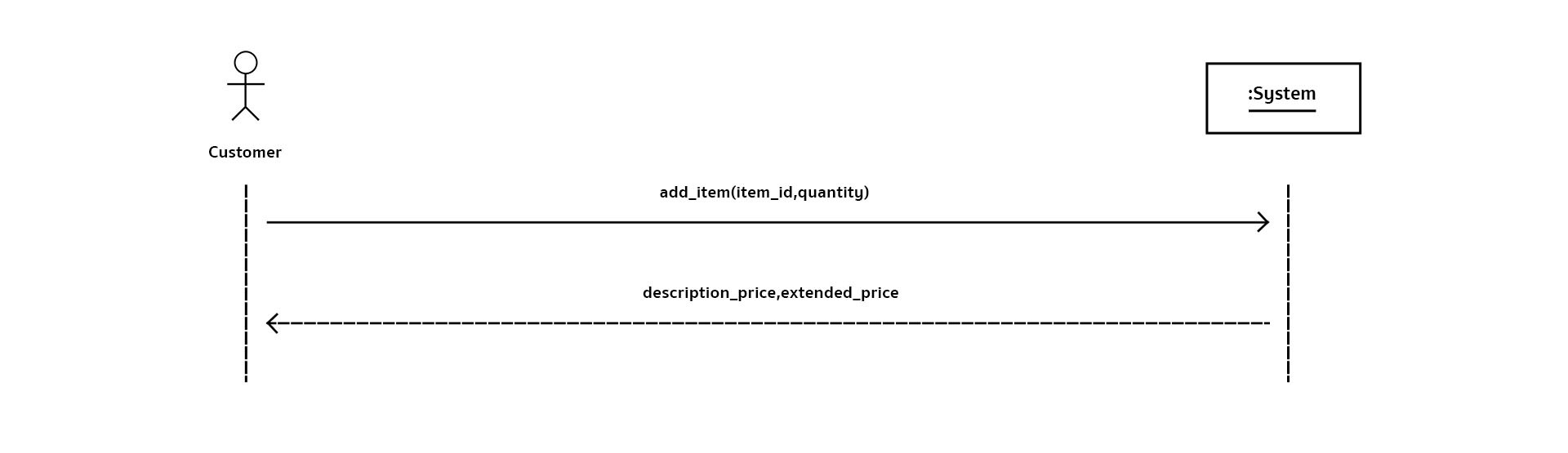
**Context diagram**

****

**DFD Level 0**

****

**2 Sequence diagram**

****

**Interface**

* **NULL**

**Testing**

* **NULL**