Warung Omega

SOFTWARE REQUIREMENTS SPECIFICATION

Agile

ONLINE RESTAURANT:
Warung Omega Indonesia

2024

0

TABLE OF CONTENTS

1

INTRODUCTION

This document includes software requirements specifications for an e-commerce website “Warung Omega”. Warung Omega is a local restaurant that sells a wide variety of seafood products, ranging from raw to processed, served with a local or localized flavor. Warung Omega aims to undergo a digital transformation by selling its products through a website. Delivering across East Jakarta, North Jakarta, West Jakarta, South Jakarta, Bekasi, and Tangerang

The purpose of this document is to give a detailed description of the requirements for “Warung Omega”. It will explain system constraints, interface and give a complete declaration for the development of the system. The website can be used by users to browse, search and add products to sell to the buyers.

Scope

Warung Omega will be an e-commerce website designed to facilitate buying of seafood products by consumers without the involvement of businesses and hidden costs. The functions that a user will be able to perform in this platform are:

- Browse through the database of available product on different category.
- View Product dashboard
- Choose as much seafood product as you can
- Payment using debit/credit card
- View receipt details
- Wait for the seafood products to get delivered

Problem

Warung Omega, a local seafood restaurant, seeks to expand its customer base and increase revenue by transitioning to a digital platform. The restaurant currently operates solely as a brick-and-mortar establishment, limiting its reach to local customers. The primary challenge lies in developing an e-commerce website that effectively showcases the restaurant's diverse seafood offerings, facilitates seamless online ordering and payment processes, and ensures timely and reliable delivery. Additionally, the website must be user-friendly and visually appealing to attract a wider customer base.

1

INTRODUCTION

People interested in buying fresh seafood products to gain super vitamin to get fit and healthy without the involvment of any middleman and free of charges are the targeted audience of this platform.

Tools to be Used:

1. Programming Language: Python, Javascript
2. Front End: Django
3. Back End: FastAPI
4. Database: Postgre / Pgadmin
5. IDE: Visual Studio Code

References

Somerville, Software Engineering, 10th ed. England: Addison-Wesley, 2017

Overview

Warung Omega wil be a B2C (Business to Customer) based web platform to selling their producs without any charges. Since, it will be designed as a web platform; it can be easily accessed by anyone with internet access and web browser in their device mobile/desktop. The advantages of our platform are:

1. Browse and search products easily
2. Add products without any charges
3. View products and its details
4. Fast-response customer service
5. Responsive design web

2

GLOSSARY

Attribute

A database attribute is a column name and the content of the fields under it in a table in a database.

B2C e-Commerce

B2C e-commerce refers to the selling of goods and services directly between a business and an individual consumer. It involves the electronic transactions conducted over the internet, allowing customers to browse, select, and purchase products or services from the comfort of their homes. Popular examples of B2C e-commerce companies include Amazon, eBay, and Alibaba.

Class

Class is a blueprint or prototype that defines the variables and the methods common to all objects of a certain kind.

FastAPI

FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.6+. Based on standard Python type hints, FastAPI allows for automatic interactive documentation, automatic data validation and serialization, and much more. It's designed to be easy to learn and use, making it a great choice for building APIs quickly and efficiently.

Interface

The interaction between a user and system running on a Web server. The user interface is the Web browser and the Web page it downloaded and rendered.

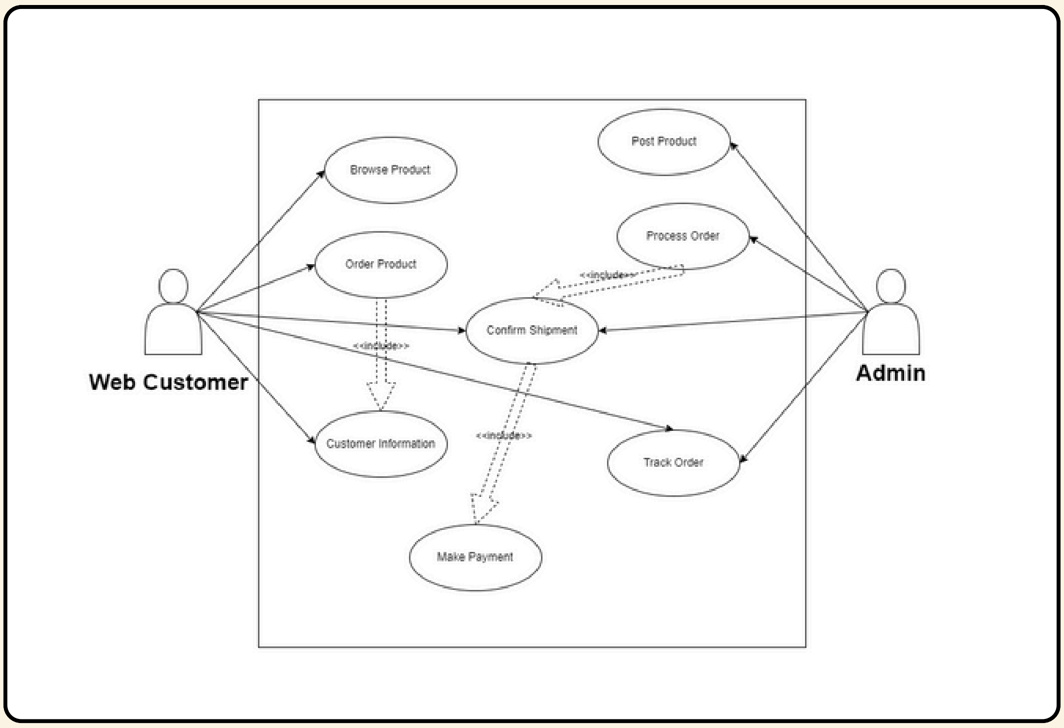
Relational Database

A relational database is any database that follows the relational model provided by traditional relational database management systems. It is a set of formally described tables from which data can be accessed or reassembled in different ways without having to reorganize the database tables.

3

SYSTEM MODEL

Use Case Diagram



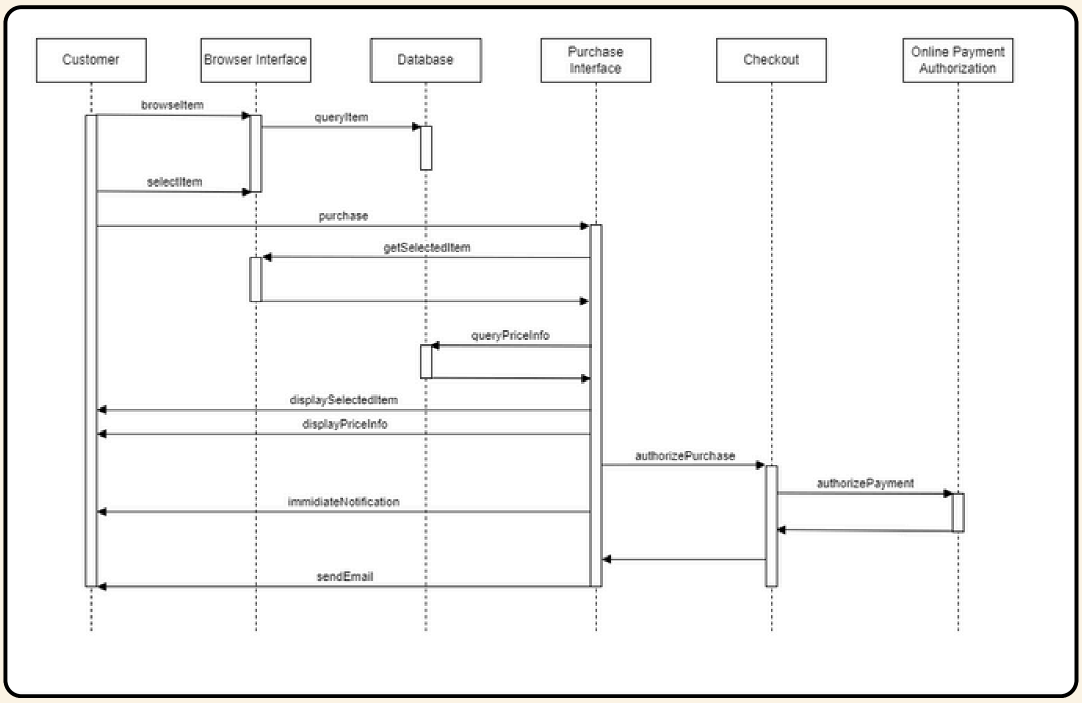
. In the "Warung Omega" e-Commerce application, there are two primary actors: the Web Customer and the Admin. The Web Customer can engage in several key activities, including browsing products, ordering products, confirming shipments, managing their information, and tracking their orders. Notably, when a Web Customer places an order, they are required to provide their personal information, as indicated by the <<include>> relationship between the "Order Product" and "Customer Information" use cases.

The Admin plays a crucial role in the system by managing product listings, processing orders, confirming shipments, and tracking orders. The Admin's interaction with the "Process Order" use case includes an <<include>> relationship with the "Confirm Shipment" use case. This relationship also involves the Web Customer, who must confirm their shipment and proceed to make a payment. The "Confirm Shipment" use case itself includes a "Make Payment" step, ensuring that transactions are completed. Both actors, the Web Customer and the Admin, interact with the "Confirm Shipment" and "Track Order" use cases, reflecting their shared responsibilities in managing and overseeing the order fulfillment process.

3

SYSTEM MODEL

Sequence Diagram

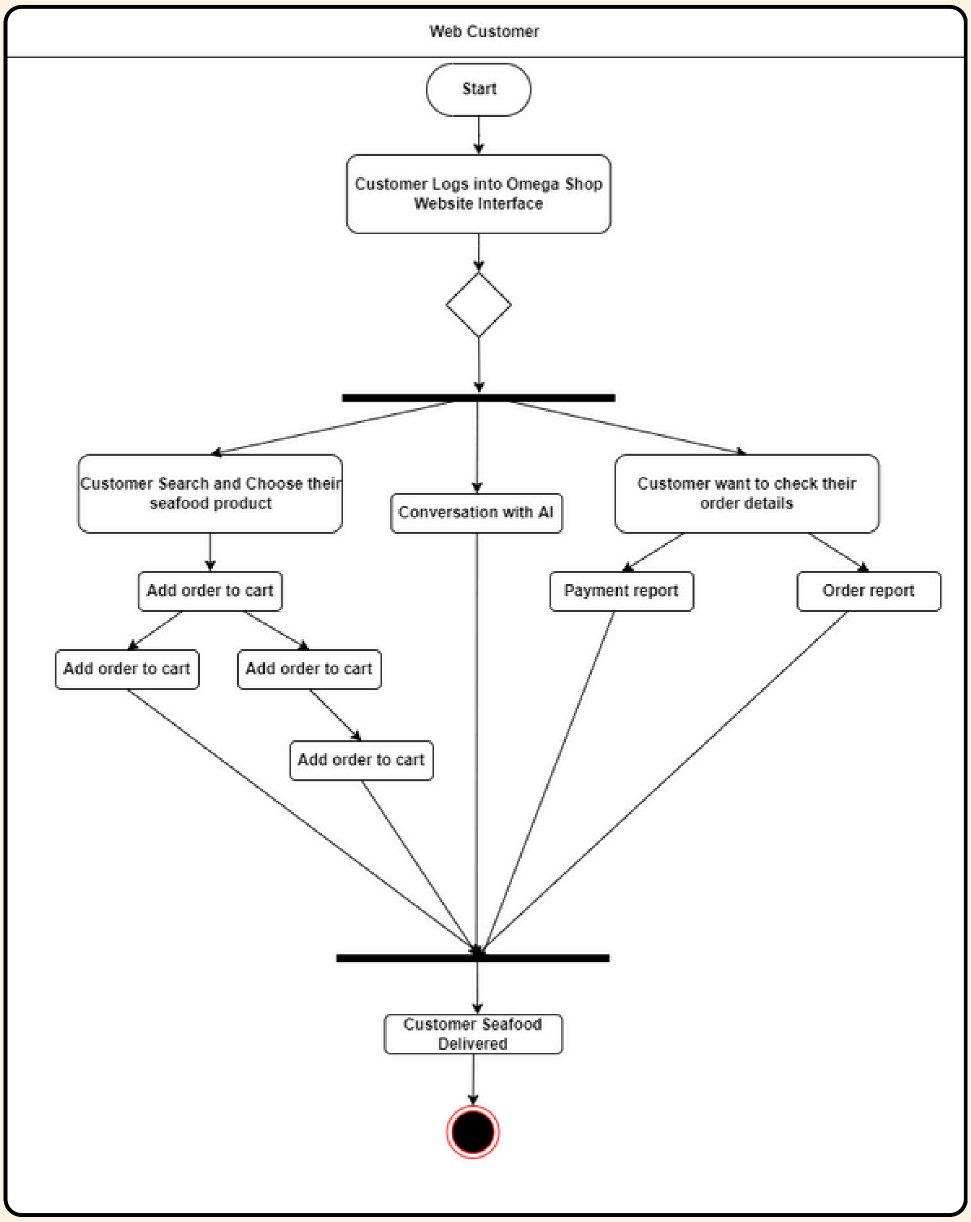


1. Customer initiates a Browse Item request to the Browser Interface.
2. The Browser Interface performs a queryItem operation on the Database to retrieve item information.
3. Customer then selects an item through the Browser Interface.
4. The Customer directly triggers a purchase action with the Purchase Interface.
5. The Purchase Interface returns the getSelectedItem details to the Browser Interface.
6. The Browser Interface relays this information back to the Purchase Interface.
7. The Purchase Interface sends a queryPriceInfo request to the Database to obtain pricing details.
8. The Database responds with the price information to the Purchase Interface.
9. The Purchase Interface then presents the displaySelectedItem and displayPriceInfo to the Customer.
10. The Purchase Interface proceeds to authorizePurchase with the Checkout component.
11. The Checkout component requests authorizePayment from the Online Payment Authorization system.
12. The Online Payment Authorization system returns the payment authorization result to the Checkout component.
13. The Checkout component sends the authorization result back to the Purchase Interface.
14. Finally, the Purchase Interface sends a sendEmail notification to the Customer confirming the purchase.

3

SYSTEM MODEL

Activity Diagram


Web Customer
Start
Customer Logs into Omega Shop Website Interface
Customer Search and Choose their seafood product
Conversation with AI
Customer want to check their order details
Add order to cart
Payment report
Order report
Add order to cart
Add order to cart
Add order to cart
Customer Seafood Delivered

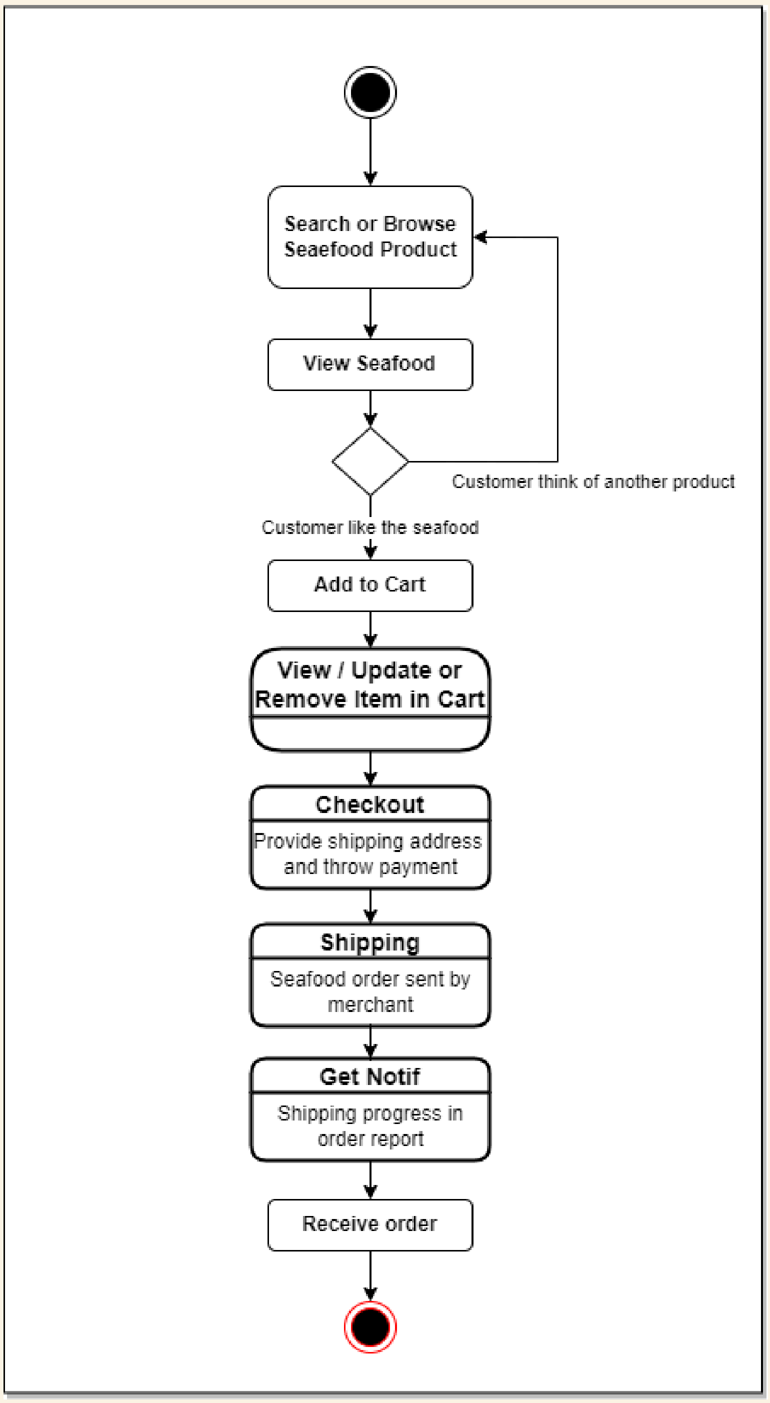
Activity diagrams are used to visualize the flow of control within a system. They are particularly helpful in understanding the sequence of activities and decision points involved in a process. Activity diagrams are often used in software development to model the workflow of a system, business process, or user interaction.

3

SYSTEM MODEL

State Chart Diagram

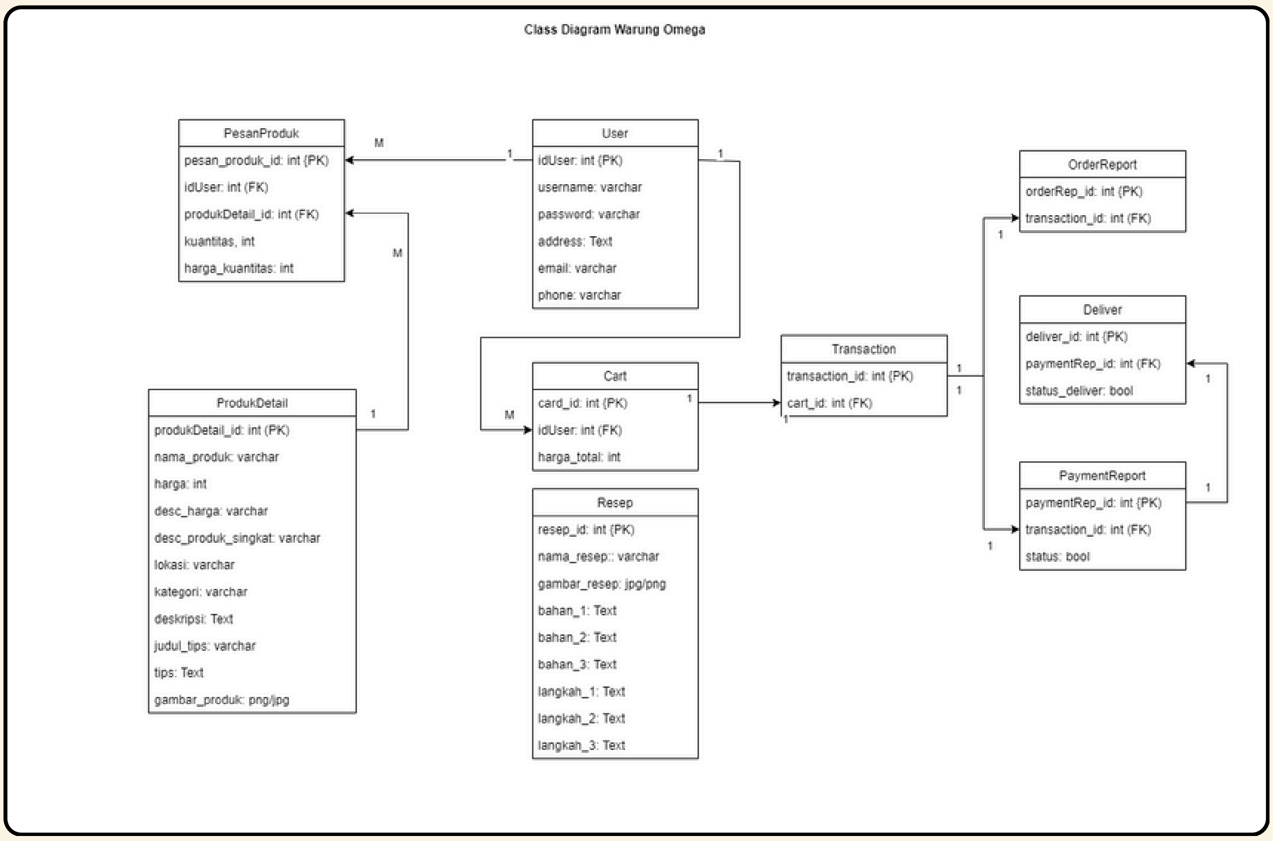
State transition diagrams are used to give an abstract description of the behavior of a system. This behavior is analyzed and represented by a series of events that can occur in one or more possible states. Here the system is started with an initial state which is black dot as shown in the above figure.


Search or Browse Seaefood Product
View Seafood
Customer think of another product
Customer like the seafood
Add to Cart
View / Update or Remove Item in Cart
Checkout
Provide shipping address and throw payment
Shipping
Seafood order sent by merchant
Get Notif
Shipping progress in order report
Receive order

3

SYSTEM MODEL

Class Diagram



In the class diagram for the "Warung Omega" e-commerce application, the User interacts with multiple components through a series of key relationships. Each User can create multiple PesanProduk entries, which represent individual product orders within their cart. Each product in ProdukDetail can be associated with numerous PesanProduk entries, detailing the product's attributes and pricing. Each User can also have multiple Cart instances, with each cart leading to a single Transaction that records the purchase. The Transaction generates a corresponding PaymentReport for payment processing and a Deliver record to track delivery status. Additionally, the Transaction is linked to an OrderReport for reporting purposes. These relationships are designed to efficiently manage product browsing, ordering, payment, and delivery processes within the application.