# Software Requirements Specification (SRS)

# Meatbox



*Baseline version 1.0*

**01082200030 - Angga Gamma Putra**

**01082200014 - Ferdynand Chandra**

**01082200001 - Rovario Khogus Limando**

**01082200031 - Gabriela Bianka**

**01082200026 – Reinhart**

**01082200018 - Dheana C Laurens**

**Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** |
| 0.1 | January 13, 2022 | All Authors | Define Product perspective, Product functions, User characteristics, General constraints, Assumptions and dependencies |
| 0.2 | January 20, 2022 | All Authors | Define Enumerated Functional Requirements, Enumerated Non-Functional Requirements and On-Screen Enumerated Requirements. |
| 0.3 | February 3, 2022 | All Authors | Defined Chapter 4 : Functional Requirement Specifications. Also added activity diagrams. |
| 0.4 | February 7, 2022 | All Authors | Redefined UseCase Diagram and Updated our Activity Diagram systematics. |
| 0.5 | March 10, 2022 | All Authors | Import Sequence Diagram |
| 0.6 | March 11. 2022 | Ferdy | Imported Matrix Retraceablity   * Class Diagram imported * Class diagram should be noted as a draft |
| 0.7 | March 30 2022 | Ferdy | Added UUCW UAW TCF EF |
| 0.8 | April 5 2022 | Ferdy | Added Use Case Test Plans  -Added comments |
| 0.9 | April 12 2022 | All authors | Updated Use Case Diagram, scenario and matrix retraceability |
| 1.0 | April 13 2022 | All authors | Updated most things  Updated Activity Diagram, and Class Diagram |

# Table of Contents

[Software Requirements Specification (SRS) for Meatbox](#_gjdgxs) 1

[Table of Contents](#_30j0zll) 4

[**CHAPTER 2**](#_1t3h5sf)8

[**2.1 Product Perspective**](#_4d34og8)8

[**2.2 Product Function**](#_2s8eyo1)8

[**2.3 User Characteristics**](#_17dp8vu)8

[**2.4 General Constraints**](#_3rdcrjn)9

[**2.5 Assumptions and Dependencies**](#_26in1rg)9

[**CHAPTER 3**](#_lnxbz9)9

[**3.1 Enumerated Functional Requirements**](#_35nkun2)9

[**3.2 Enumerated Non-Functional Requirements**](#_1ksv4uv)10

[**3.3 On-Screen Enumerated Requirements**](#_44sinio)10

[**CHAPTER 4**](#_9leubx24ey2r)11

[4.1 Stakeholders and Goals](#_n5bd0ugqckjr) 11

[4.2 Actors and Goals](#_zcu7vm7hjb6t) 12

[4.3 Use Case](#_6ymlyz3sk7gq) 13

[4.4 Use Case Scenario](#_6ygajdeo3jx3) 14

[4.5 Activity Diagam](#_k82ybx25r11r) 20

[4.6 Sequence Diagram](#_b59vm1nu5mit) 27

[4.6.1 UC-01 Browse Product](#_z337ya) 27

[4.6.2 UC-02 Make Purchase](#_nc55oxe56g8h) 27

[4.7 Matrix Retraceablity](#_pvb5mdz9rzll) 28

[4.8 Class Diagram](#_2nhhqtezdpb2) 29

[4.9 Use Case Test Plans](#_cqz4v59vxcxb) 29

[**CHAPTER 5**](#_z8h4tfo01k27)34

[5.1 Use Case Weight](#_wz9a7z89kdr6) 34

[5.1.1 Unadjusted Use Case Weight](#_7b1dpccmkfwt) 34

[5.1.2 Unadjusted Actor Weight](#_nqsg3qbbjreg) 34

[5.1.3 Technical Complexity Factor](#_bf2k16kjbu4t) 35

[5.14 Enviromental Complexity Factor](#_o4x8a1x41yq5) 36

[5.1.5 Use Case Point Calculation](#_ng6mjsic9sex) 37

# **CHAPTER 1**

**INTRODUCTION**

## **1.1 Purpose**

The purpose of this document is to present a detailed description of the MeatBox. It will contain an overall description and some functionally oriented features and a few functional requirements and specifications for parts of the software. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

## **1.2 Scope**

• Simpler and save customer time. The customer simply inputs the part of the meat/goods needed and how much is needed and the order will be packaged immediately.

• More efficient because the process is automated

• Reducing expenses in the form of workers' salaries, because the machine will carry out orders with input from customers

• Customer Satisfaction - Because all processes are carried out by the customer, the results obtained will be more or less in accordance with what the customer wants.

• This machine runs in stages, when there is an error, it is easier to track and repair

• Saves company time because there is no need to do worker training Quality control is safer because of machine work

• Self Order : Self order easily by customer without waiter.

• Strong discount schemes: make it easy to offer discounts or promotions to attract more customers to the restaurant

• cash flow: it is easier for us (restaurant owners) to manage finances

## **1.3 Definitions, Acronyms, and Abbreviations**

|  |  |  |
| --- | --- | --- |
| **Term/Acronym** | **Definition** | **Description** |
| SRS | Software Requirement Specification | This document |
| UC | Use Case | A numbered code(UC-XX) to uniquely identify use case |
| REQ | Requirements | A numbered code (REQ-XX) to uniquely identify functional, non-functional, and enumerated on-screen requirements |
| TC | Test Case | A numbered code (TC-XX) to uniquely identify use case test plans |

## **1.4 Overview**

This document will provide a general overview of the system being made. It covers none other than the system’s functions and constraints, non-functional requirements, functional requirements, use cases, test plans, and project plan. The following section of this document will focus on describing the system in terms of product perspective, product functions, user characteristics, assumptions and dependencies. In the third section, we will address specific requirements of the system, which will enclose functional and nonfunctional requirements, and also classes and objects within the system. The fourth section will cover the functional requirements in more detail, and the fifth section will briefly explain the project management plan.

# **CHAPTER 2**

**GENERAL DESCRIPTION**

## **2.1 Product Perspective**

Introducing ***Meatbox, Meatbox*** is a fully automated butchery shop that serves a wide variety of meat products. Ranging from cows, pigs, fish, chickens to even sea stars like sea urchins. We strive and aim to apply perfection and personalization to our products. As a customer myself who always bought meat from the butcher shops, they never made the cuts even or sometimes they just forgot to process our order due to the number of people requesting their service. That causes us to think about how we can fix this. How we can fix the community of butcher shops. Or you can just be that one shy person that is way too shy to even call the butcher. Well, fret no more. Because ***Meatbox*** is the solution for all of us.

## **2.2 Product Function**

The system will perform and/or provide the following functions:

1. Effective for customers who wanted to shop their personalized meat and poultry products by giving the customer the freedom of choice upon processing their products
2. You don’t need to bring any cash anymore to pay the butcher because we made a replacement for the traditional payment system. Because we will accept online payment .wallets such as Gopay, OVO, to even the legendary Bitcoin!
3. Shop on the go: Gone are the days of spending more time describing how you wanted your meat to be chopped or how you wanted the fish to be cut. With this machine, you can get it however you wanted it to be and save more time because you don’t need to explain every time!

## **2.3 User Characteristics**

The system will require users to have the following qualifications:

1. Customers are expected to be technology literate to be able to fully take advantage of the system. The ***Meatbox*** will have a search bar located at the monitor of the machine to check a certain item’s availability. The payment must be carried out by using some form of online payment method.
2. ***Meatbox*** employees are expected to have a degree of intelligence to be able to service the machine. Do not worry because the machine will tell them exactly what needs to be done for the machine

## **2.4 General Constraints**

The system will have the following constraints when delivering the software:

1. Gather user personalization data to be able to provide accurate recreation of the desired product.
2. A large amount of data is needed to record each item and function within and its availability in the ***Meatbox***, which requires a considerate amount of disk space.
3. Gathering user data in real-time requires the application to be reliable and safe to carry out this task. The system must be able to produce new data depending on user behavior and provide security towards user data.
4. Fast and reliable internet connection. As the system needs an internet connection to function, a stable and quick internet connection is critical.

## **2.5 Assumptions and Dependencies**

This system depends on the work of the machine and online wallet payment service.

Features included in the system needed requirements such as database management tools, user data, item data, mobile navigation system, the user needs a camera-equipped mobile device, and more. These requirements are necessary for the system to be able to fully function. This system will also strongly depend on an internet connection to run since it requires online payment to confirm purchases.

# **CHAPTER 3**

**INTERNAL STRUCTURE**

## **3.1 Enumerated Functional Requirements**

|  |  |
| --- | --- |
| REQ-01 | System shall allow customer to input the products the user wanted to be searched upon the catalog of products |
| REQ-02 | System shall allow customer to choose the payment method consisting of QR scan using online wallet services. |
| REQ-03 | System shall allow customer to check the ready stock of a certain item |
| REQ-04 | System shall allow Customer to add item to their virtual cart |
| REQ-05 | System shall allow customer to add multiple items to their virtual cart |
| REQ-06 | System shall allow customer to remove item(s) from their virtual cart. |
| REQ-07 | System shall allow customer to scan payment Quick Response (QR) Code |
| REQ-08 | System shall allow Customer to pay for items in their virtual cart via an online payment method |
| REQ-09 | System shall allow Inventory Manager to update information on items |
| REQ-10 | The system will allow customers to cancel scanning the Quick Response Code (QR) payment when an issue occurs |

## **3.2 Enumerated Non-Functional Requirements**

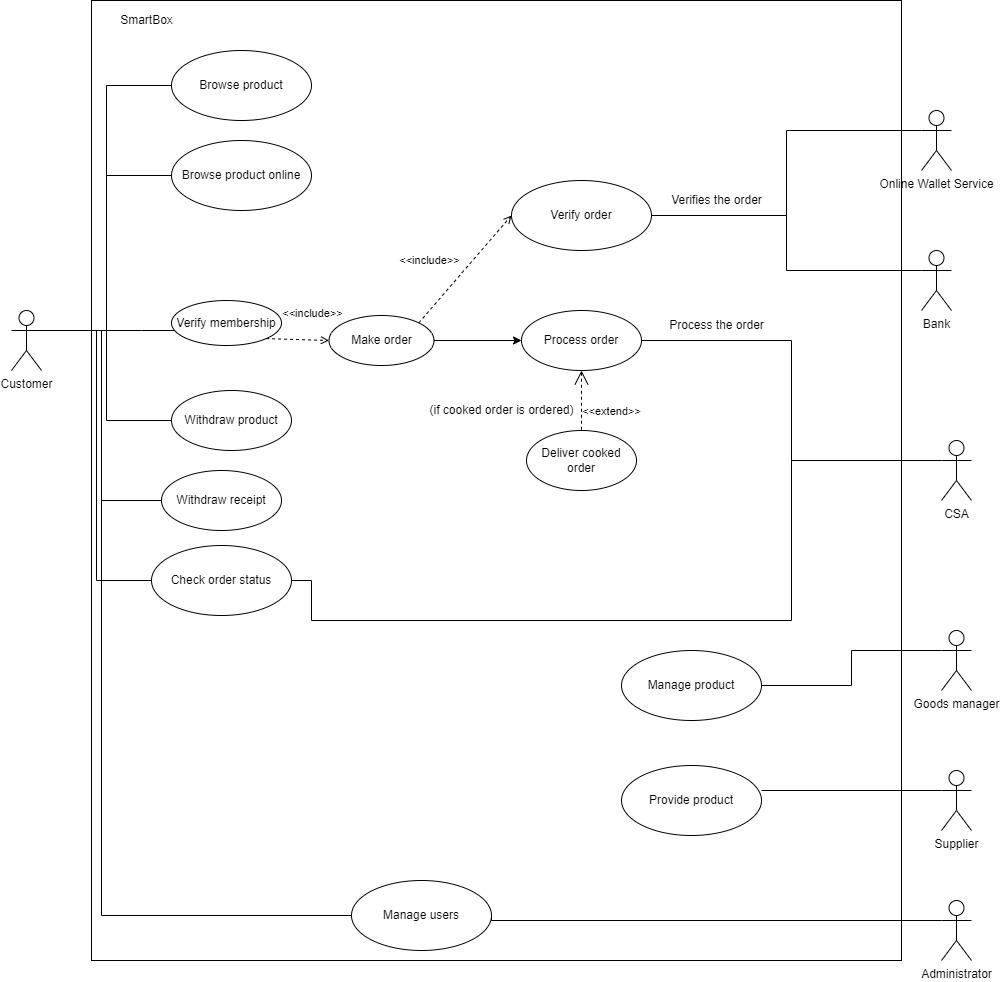
|  |  |
| --- | --- |
| REQ-11 | USER is allowed to take the item by opening the take item chamber |
| REQ-12 | System shall allow the application to function through internet connection speeds of at least 1 Mbps |
| REQ-13 | System shall recommend items that are on sale to Customer |
| REQ-14 | Customers are given a limited amount of time to pay for their purchases |

## **3.3 On-Screen Enumerated Requirements**

|  |  |
| --- | --- |
| REQ-15 | System is designed to be connected to the internet for stockkeeping |
| REQ-16 | System shall use English and bahasa Indonesia as the application interface language |
| REQ-17 | System shall have an intuitive interface that supports touch screen operations |
| REQ-18 | System shall have a dark themed interface to protect eyesight |

# **CHAPTER 4**

**FUNCTIONAL REQUIREMENT SPECIFICATION**

****

## **4.1 Stakeholders and Goals**

|  |  |
| --- | --- |
| **Stakeholders** | **Goals** |
| Meat Shop | to make customers more comfortable in ordering meat so that the goods needed are quickly taken by customers by providing self service so that customers can choose meat. |
| Customer | make it easier for customers to make payments and minimize time for packing goods |
| Supplier | for the stock of meat in the Meat shop. |

## 

|  |  |
| --- | --- |
| Online Wallet Service | To make the customer able to pay using online wallets such as gopay and ovos |

|  |  |
| --- | --- |
| **CSA** | For assisting the users incase |
| Administrator | Helping to manage the users using human brain power capability |
| Bank | To make inclusions of finance related issues |
| Goods Manager | Responsible for managing the product |

## **4.2 Actors and Goals**

|  |  |
| --- | --- |
| **Actors** | **Goals** |
| Customers | customers can search and determine the goods they will buy. After that, the customer confirms the payment and uses the payment method. Customers can wait for their orders. |
| Supplier | to order items for sale at the butcher shop |
| Online Wallet Service | Once the online wallet is authenticated, the purchase is approved or denied, the money is added to the customer's wallet bill, and the payment is credited to the merchant's account. |

|  |  |
| --- | --- |
| **CSA** | Costumer Service Assistant, responsible for processing order from customer. CSA assigns chef to process the product or check the order status of the currently processing order by the User. |
| Administrator | Responsible for managing the users. |
| Bank | Responsible for the main objectives of setting up of payments banks are to ensure the financial inclusion |
| Goods Manager | Responsible for managing the product |

## **4.3 Use Case**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | **Description** |
| UC-1 | **Browse Product** | Customer browses the catalog of products by interacting with the interfaces of our MeatBox system. Customers are presented with a variety of products. Here customers will choose their product and then they will |
| UC-2 | **Make Purchase** | Allows the customer to make purchase by paying for the product using an online wallet service and then confirmed by the Online wallet service after making the payment. The machine will also print out receipt along with dispensing the product. |
| UC-3 | **Withdraw Product** | Allows the customer to withdraw any product from the machine’s dispenser box **that has been dispensed, regardless of who bought it.** |
| UC-4 | **Withdraw Receipt** | Allows the customer to withdraw receipt from the receipt dispenser **after it has been dispensed.** |
| UC-5 | **Provide product** | Allows the supplier to make purchase for the machine, incase of item shortage (this can be seen on the machine because the machine has an automatic low supply indicator) |
| UC-6 | **Verify membership** | The machine authenticates the user to define if they are eligible to use the machine. |
| UC-7 | **Make Order** | Allows the user to make an order according to their wishes on the machine |
| UC-8 | **Process Order** | The machine processes the order accordingly |
| UC-9 | **Deliver cooked order** | The machine tells the employee to order the cooked order if the user wants it to be cooked |
| UC-10 | **Verify order** | The machine verifies the order |
| UC-11 | **Manage Product** | This allows eligible user to manage the products |
| UC-12 | **Manage users** | This allows the actor to manage the user if an admin was using this, then they can manage all user |
| UC-13 | **Check Order Status** | This allows the CSA to check the order status if the customer request an assistance about their order (including issues) |

## **4.4 Use Case Scenario**

|  |  |
| --- | --- |
| Use Case Name | Browse Product |
| Use Case ID | UC-01 |
| Actor’s Goal | To know the product in shop |
| Primary Actor | Customer |
| Participating Actors | Customer |
| Preconditions | Customer not yet obtain the product information |
| Postconditions | Customer obtains the product information |
| Main Success Scenario | 1. Customer search for the product information 2. System request filter for product name, size, price, weight 3. Customer insert product name, size, price, weight 4. System display the filtered product name, size, price, weight 5. Customer get the product information |

|  |  |
| --- | --- |
| Use Case Name | Make Purchase |
| Use Case ID | UC-02 |
| Actor’s Goal | To purchase the products that customer chose |
| Primary Actor | Customer |
| Participating Actors | Customer |
| Preconditions | Customer has chosen the products |
| Postconditions | Customer can choose to confirm payment or not |
| Main Success Scenario | 1. Customer search the product 2. The customer chose the product 3. Customer adds the product to shopping cart 4. The customer decides product quantity and gives additional notes 5. Customer checks out |

|  |  |
| --- | --- |
| Use Case Name | Withdraw product |
| Use Case ID | UC-03 |
| Actor’s Goal | To complete purchase |
| Primary Actor | Customer |
| Participating Actors | Online Wallet Payment Service |
| Preconditions | Customer has confirm payment |
| Postconditions | Customer completes the purchase |
| Main Success Scenario | 1. Customer make payment 2. System process the payment 3. E-wallet payment service confirm transaction 4. Customer confirm the payment 5. Customer get the product |

|  |  |
| --- | --- |
| Use Case Name | Withdraw receipt |
| Use Case ID | UC-04 |
| Actor’s Goal | To complete purchase |
| Primary Actor | Customer |
| Participating Actors | Online Wallet Payment Service |
| Preconditions | Customer has confirm payment |
| Postconditions | Customer completes the purchase |
| Main Success Scenario | 1. Customer confirm the payment 2. Customer get the product |

|  |  |
| --- | --- |
| Use Case Name | Verify Membership |
| Use Case ID | UC-5 |
| Actor’s Goal | To complete purchase |
| Primary Actor | Customer |
| Participating Actors | Customer |
| Preconditions | The Machine ensures that the user is qualified to use the machine |
| Postconditions | The Machine has authenticated the user as an eligible member of Smartbox. |
| Main Success Scenario | 1. Customer has logged in |

|  |  |
| --- | --- |
| Use Case Name | Provide supply |
| Use Case ID | UC-06 |
| Actor’s Goal | To complete purchase |
| Primary Actor | Supplier |
| Participating Actors | Supplier |
| Preconditions | The Machine’s Low Supply Indicator has turned on. |
| Postconditions | The Machine’s supplies has been supplied. |
| Main Success Scenario | 1. Supplier resupply the machine. 2. Machine’s Supplies are resupplied |

|  |  |
| --- | --- |
| Use Case Name | Make Order |
| Use Case ID | UC-7 |
| Actor’s Goal | To make order in the machine |
| Primary Actor | Customer |
| Participating Actors | Customer |
| Preconditions | Customer is choosing the product |
| Postconditions | The customer has chosen the product and has made an order |
| Main Success Scenario | * Customer has chosen the product * The Machine process the product chosen |

|  |  |
| --- | --- |
| Use Case Name | Process Order |
| Use Case ID | UC-8 |
| Actor’s Goal | The Machine processing the message order from customer |
| Primary Actor | CSA |
| Participating Actors | CSA, Customer |
| Preconditions | Customer orders product to be processed |
| Postconditions | The Machine has processed customer orders and is in the packaging stage |
| Main Success Scenario | * The machine can process orders |

|  |  |
| --- | --- |
| Use Case Name | Deliver Cooked Order |
| Use Case ID | UC-9 |
| Actor’s Goal | The Machine notifies employees to cook orders from customers |
| Primary Actor | CSA |
| Participating Actors | CSA |
| Preconditions | Customers can choose whether the food can be cooked by the chef or those who cook it. |
| Postconditions | Chef get the notification order from the Machine |
| Main Success Scenario | * Raw food has been processed |

|  |  |
| --- | --- |
| Use Case Name | Verify Order |
| Use Case ID | UC-10 |
| Actor’s Goal | The Machine verifies orders from customers |
| Primary Actor | Online Wallet Service |
| Participating Actors | Online Wallet Service |
| Preconditions | The Machine asks for verification from OWS |
| Postconditions | OWS confirmed the verification |
| Main Success Scenario | * Order has been verified |

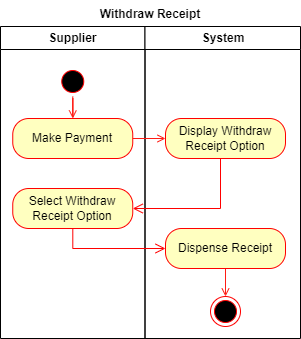
|  |  |
| --- | --- |
| Use Case Name | Manage Product |
| Use Case ID | UC-11 |
| Actor’s Goal | Machine allows the actor to manage the products |
| Primary Actor | Goods Managers |
| Participating Actors | Goods Managers |
| Preconditions | The Machine ask the customer from ensure users to be qualified to manage the product |
| Postconditions | The product has been managed. |
| Main Success Scenario | * Goods has been replenished |

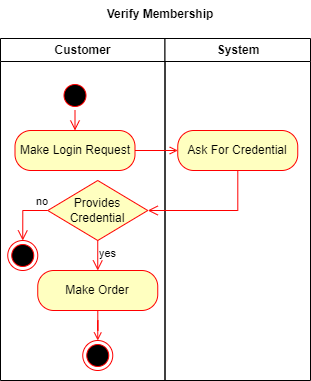
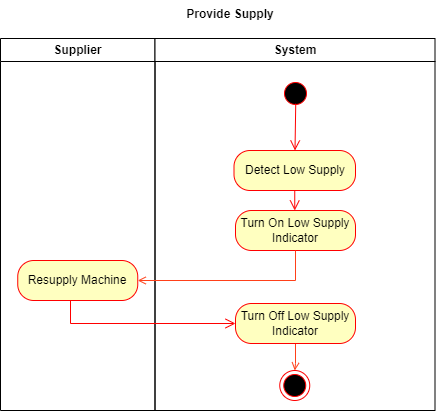
|  |  |
| --- | --- |
| Use Case Name | Manage Users |
| Use Case ID | UC-12 |
| Actor’s Goal | To manage the user allowing admins to manage all user and customer to only manage their own user account |
| Primary Actor | Administrator |
| Participating Actors | Administrator, customer |
| Preconditions | The user has not been managed |
| Postconditions | The user has been managed |
| Main Success Scenario | * User has created their account * User has logged in to their account |

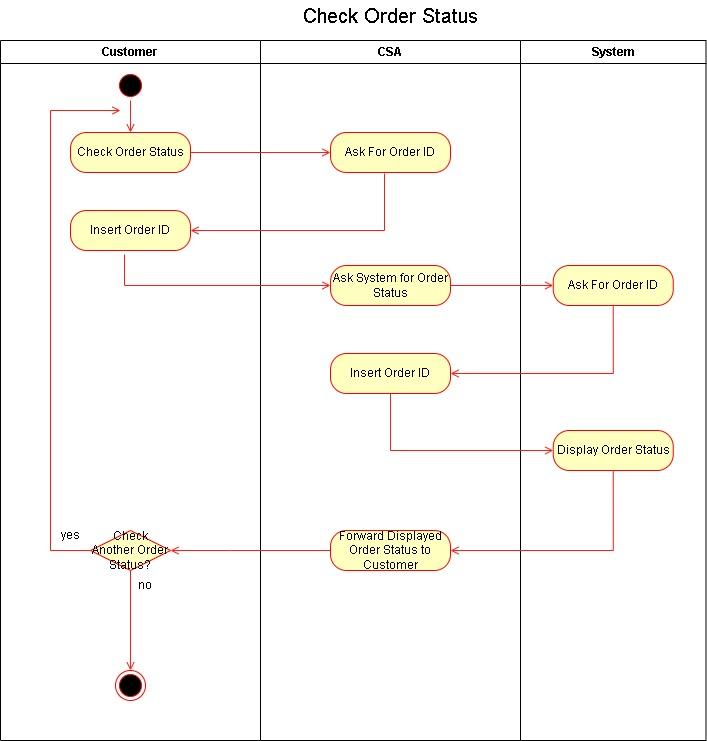
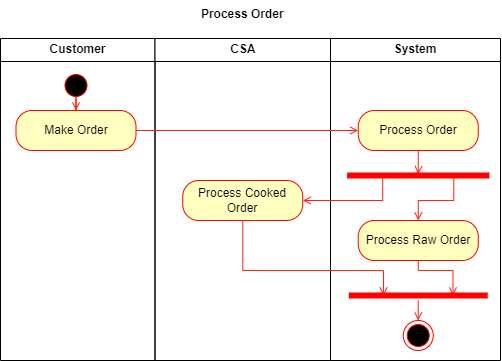
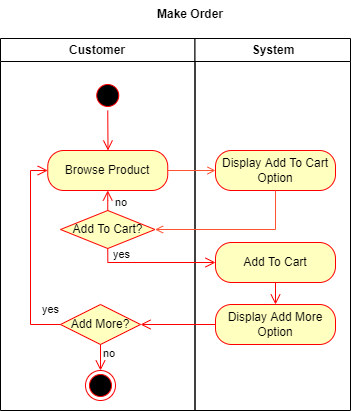
|  |  |
| --- | --- |
| Use Case Name | Check Order Status |
| Use Case ID | UC-13 |
| Actor’s Goal | The Machine help to check their order if there is a problem with their order |
| Primary Actor | CSA |
| Participating Actors | CSA, Customer |
| Preconditions | The Customer can see their order status and ask for help if their have the request about their order |
| Postconditions | CSA has verified the issue with the system |
| Main Success Scenario | * CSA has checked the order status and verifies the issue |

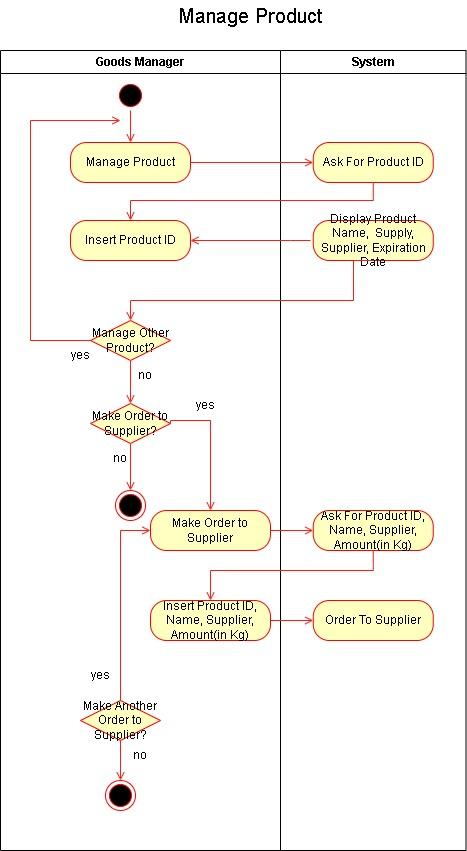
## **4.5 Activity Diagram**

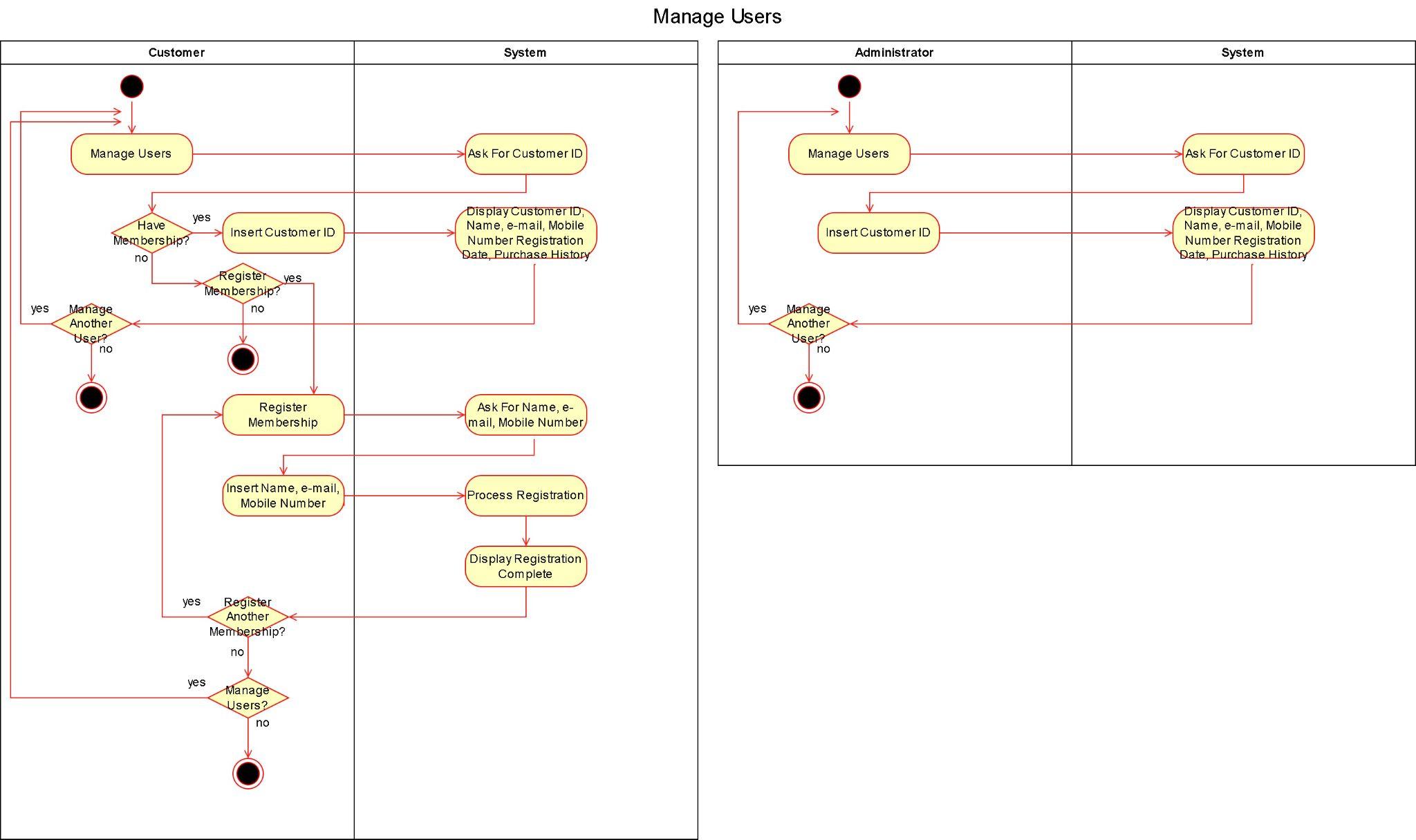
## 

****

****

****

****



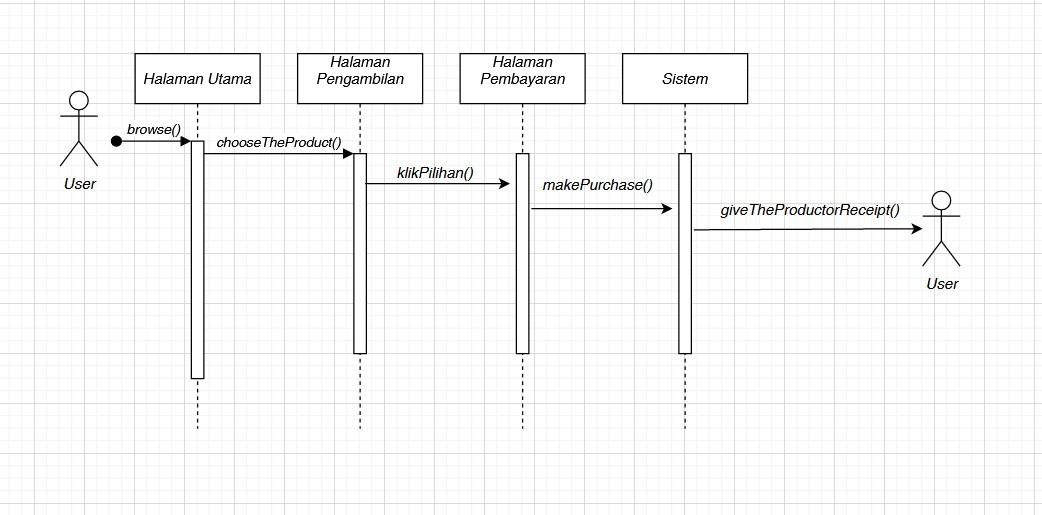
## **4.6 Sequence Diagram**

### **4.6.1 UC-01 Browse Product**

Diagram

Description automatically generated

### **4.6.2 UC-02 Make Purchase**

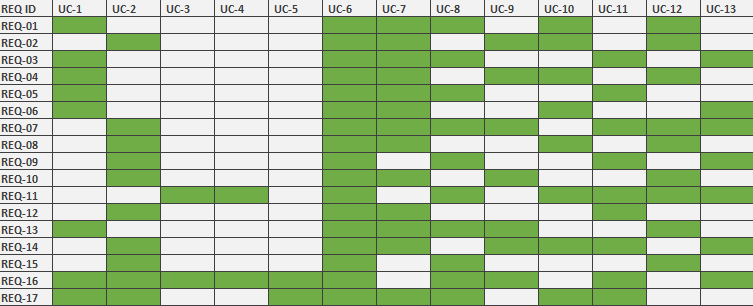


## 

## 

## 

## **4.7 Matrix Retraceablity**



## **4.8 Class Diagram**

Diagram

Description automatically generated

## **4.9 Use Case Test Plans**

|  |  |  |  |
| --- | --- | --- | --- |
| Test case ID | TC-1 | | |
| Test case name | Browse Product – Normal Flow | | |
| Preconditions | Customer requests to browse the products | | |
| Action | | Expected Results | Test Data |
| 1. Tap on Browse Product button | | System displays list of products |  |
|  | |  |  |
|  | |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Test case ID | TC-2 | | |
| Test case name | Browse Product – Valid Search | | |
| Preconditions | Customer requests to browse the products | | |
| Action | | Expected Results | Test Data |
| 1. Tap on Browse Product button | | System displays list of products |  |
| 2.Tap on Search Product button | | System display an input bar and a virtual keyboard for customers. |  |
| 3.Input a product ID/Name | | System search the stocks of the inputted value. | Wagyu A5 |
| 4.Tap on Enter button | | System displays the product searched |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Test case ID | TC-3 | | |
| Test case name | Browse Product – Invalid Search | | |
| Preconditions | Customer requests to browse the products | | |
| Action | | Expected Results | Test Data |
| 1. Repeat Action 1-3 of TC-3 | | System displays list of products | Kursi Gaming |
| 2. Tap on Enter button | | System displays nothing |  |
|  | |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Test case ID | TC-4 | | |
| Test case name | Make Purchase – Normal Flow | | |
| Preconditions | Customer request to make purchase of the product selected | | |
| Action | | Expected Results | Test Data |
| 1.Customer insert product into the ‘shopping cart’ | | System insert product value to the cart system. | 1 Raw tenderloin, 3 Custom sliced turkey |
| 2.Customer proceed to the checkout button | | System displays the product that is going to be checked out and the total of the payment |  |
| 3.Customer input payment method | | System displays a list of payment method | OVO, 085200120045, 772233 |
| 4.Customer tap on ‘pay’ button | | System validates the payment request and generates confirmation |  |
| 5. Customer tap on ‘process’ button | | System processes the request. |  |

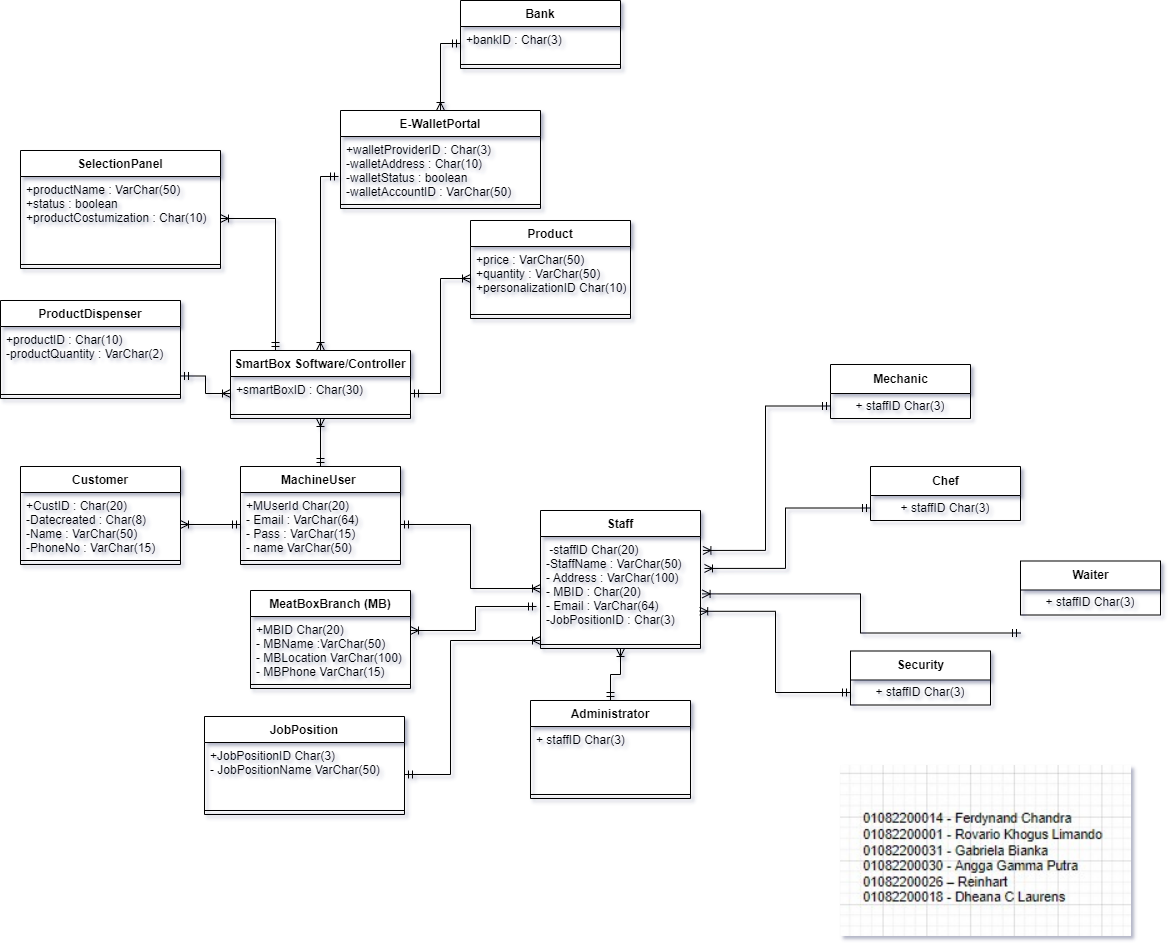
|  |  |  |  |
| --- | --- | --- | --- |
| Test case ID | TC-5 | | |
| Test case name | Make Purchase – Invalid Payment | | |
| Preconditions | Customer request to make purchase of the product selected | | |
| Action | | Expected Results | Test Data |
| 1.Repeat action 1-3 of TC-4 | |  | - |
| 2.Customer tap on pay button | | System displays an error message |  |
|  | |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Test case ID | TC-6 | | |
| Test case name | Provide supply – Normal Flow | | |
| Preconditions | Admin request to make supply purchases for the machine | | |
| Action | | Expected Results | Test Data |
| 1. Admin tap on Administrator mode | | System display the list of option available for administrator. |  |
| 2.Admin tap on Provide supply button | | System display the list of supplies that can be purchased from the official stockist |  |
| 3. Admin input the products that is going to be purchased | | System lists the product input | 100 Raw Meat |
| 4.Admin insert the order from the machine | | System insert the product into the system Admin Shopping Cart |  |
| 5. Admin tap on the process button | | System processes the order that is made by the Admin |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Test case ID | TC-7 | | |
| Test case name | Provide supply – Invalid product search | | |
| Preconditions | Admin request to make supply purchases for the machine | | |
| Action | | Expected Results | Test Data |
| 1.Repeat action 1-2 | |  |  |
| 2. Admin input the products that is going to be purchased | | System lists the product input | Product List |
| 3.Admin insert the order from the machine | | System display error |  |
| 4.Admin makes the order from the machine | | System display an error message |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Test case ID | TC-X | | |
| Test case name | Verify order – Normal flow | | |
| Preconditions | Customer ordered product | | |
| Action | | Expected Results | Test Data |
| 1.Repeat action 1-2 | |  |  |
| 2. Admin input the products that is going to be purchased | | System lists the product input | Product List |
| 3.Admin insert the order from the machine | | System display error |  |
| 4.Admin makes the order from the machine | | System display an error message |  |

**4.10 TRD**

****

# CHAPTER 5

**RENCANA MANAJEMEN PROYEK**

## **5.1 Use Case Weight**

Use case points (UCP or UCPs) is a software estimation technique used to forecast the software size for software development projects. UCP is used when the Unified Modeling Language (UML) and Rational Unified Process (RUP) methodologies are being used for the software design and development. The concept of UCP is based on the requirements for the system being written using use cases, which is part of the UML set of modeling techniques. The software size (UCP) is calculated based on elements of the system use cases with factoring to account for technical and environmental considerations. The UCP for a project can then be used to calculate the estimated effort for a project.

**Source :** [Use case points - Wikipedia](https://en.wikipedia.org/wiki/Use_case_points#:~:text=Unadjusted%20Use%20Case%20Weight%20(UUCW)%20%E2%80%93%20the%20point%20size%20of,number%20and%20complexity%20of%20actors.)

### **5.1.1 Unadjusted Use Case Weight**

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | Weight | Number of Use Cases | Product |
| Simple | 5 | 2 | 10 |
| Average | 10 | 1 | 10 |
| Complex | 15 | 2 | 30 |
| Total |  |  | 50 |

### **5.1.2 Unadjusted Actor Weight**

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | Weight | Number of Use Cases | Product |
| Simple | 1 | 1 | 1 |
| Average | 2 | 1 | 2 |
| Complex | 3 | 1 | 3 |
| Total |  |  | 6 |

### **5.1.3 Technical Complexity Factor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Factor** | **Description** | **Weight** | **Assigned Value** | **Weight x Assigned Value** |
| T1 | Distributed system | 2.0 | 5 | 10 |
| T2 | Response time/performance objectives | 1.0 | 5 | 5 |
| T3 | End-user efficiency | 1.0 | 3 | 3 |
| T4 | Internal processing complexity | 1.0 | 2 | 2 |
| T5 | Code reusability | 1.0 | 3 | 3 |
| T6 | Easy to install | 0.5 | 1 | 0.5 |
| T7 | Easy to use | 0.5 | 5 | 2.5 |
| T8 | Portability to other platforms | 2.0 | 2 | 4 |
| T9 | System maintenance | 1.0 | 2 | 2 |
| T10 | Concurrent/parallel processing | 1.0 | 3 | 3 |
| T11 | Security features | 1.0 | 5 | 5 |
| T12 | Access for third parties | 1.0 | 1 | 1 |
| T13 | End user training | 1.0 | 1 | 1 |
| **Total (TF):** | | | | **42** |

TCF = 0.6 + (42/100) = 1.02

TCF = 1.02

### **5.14 Enviromental Complexity Factor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Factor** | **Description** | **Weight** | **Assigned Value** | **Weight x Assigned Value** |
| E1 | Familiarity with development process used | 1.5 | 3 | 4.5 |
| E2 | Application experience | 0.5 | 3 | 1.5 |
| E3 | Object-oriented experience of team | 1.0 | 2 | 2 |
| E4 | Lead analyst capability | 0.5 | 5 | 2.5 |
| E5 | Motivation of the team | 1.0 | 2 | 2 |
| E6 | Stability of requirements | 2.0 | 1 | 2 |
| E7 | Part-time staff | -1.0 | 0 | 0 |
| E8 | Difficult programming language | -1.0 | 4 | -4 |
| **Total (EF):** | | | | **10.5** |

ECF = 1.4 + (-0.03 \* 10.5) = 1.085

ECF = 1.085

### 5.1.5 Use Case Point Calculation

UCP = (UUCW + UAW) x TCF x ECF

UCP = (50 + 6) x 1.02 x 1.085

UCP = 61.9

Estimated Effort = UCP x Hours/UCP

Estimated Effort = 61.9 \* 28 = 1,733.2 Hours