SKPL - OO - 01

SOFTWARE REQUIREMENT SPECIFICATION

**Food Ordering Mobile Application for Small-scale Restaurant**

for:

Student and Lecturer at Telkom University

Prepared by:

- Raka Adhi Bagaskara (1301164587)

- M. Rikbal Ikhsani (1301163598)

- Hasna F Hasya (1301164594)

- Sudyatmika Putra (1301180494)

Bachelor of Informatics – Faculty of Informatics (School of Computing)

Telkom University

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

|  |  |  |  |
| --- | --- | --- | --- |
| Informatics Bachelor-  Faculty of Informatics | Document Number | | Pages |
| *SKPL-OO-01* | |  |
| Revision |  | *Date:* |

# Changes List

|  |  |
| --- | --- |
| Revision | Description |
| A | Purpose of writing the document, Document coverage, Definition, Abbreviation, and Acronym, Reference |
| B | Input the flowchart, edit the use case |
| C | Edit the number, add list of figure and list of table, insert the table and the figure |
| D | Edit the Class Diagram, edit the Use case Diagram, input the use case diagram 13 |
| E |  |
| F |  |
| G |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | - | A | B | C | D | E | F | G |
| DATE |  |  |  |  |  |  |  |  |
| Written by |  |  |  |  |  |  |  |  |
| Examined by |  |  |  |  |  |  |  |  |
| Approved by |  |  |  |  |  |  |  |  |

# Pages Changes List

|  |  |  |  |
| --- | --- | --- | --- |
| Page | Revision | Page | Revision |
|  |  |  |  |

# Table of Contents

[Changes List 2](#_Toc39425684)

[Pages Changes List 1](#_Toc39425685)

[Table of Contents 3](#_Toc39425686)

[List of Tables 5](#_Toc39425687)

[List of Figures 6](#_Toc39425688)

[1. Preliminary 7](#_Toc39425689)

[1.1 Purpose of Writing Document 7](#_Toc39425690)

[1.2 Document Coverage 7](#_Toc39425691)

[1.3 Definition, Abbreviation, and Acronym 7](#_Toc39425692)

[1.4 References 7](#_Toc39425693)

[2. Software Global Description 8](#_Toc39425694)

[2.1 Software Statement of Objective 8](#_Toc39425695)

[2.2 Software Perspective and Function 8](#_Toc39425696)

[2.3 User Profile and Characteristic 8](#_Toc39425697)

[2.4 Operating Environment 9](#_Toc39425698)

[2.5 Software/system Boundaries 9](#_Toc39425699)

[2.6 Assumption and Dependency 9](#_Toc39425700)

[3. Software Detailed Description 10](#_Toc39425701)

[3.1 Requirement Description 10](#_Toc39425702)

[3.1.1 Functional Requirement 10](#_Toc39425703)

[3.1.2 Non-Functional Requirement 11](#_Toc39425704)

[3.2 Analytical Modelling 12](#_Toc39425705)

[3.2.1 Usecase Diagram 12](#_Toc39425706)

[**3.2.1.1** **Use case Scenario #1** 13](#_Toc39425707)

[**3.2.1.2** **Use case Scenario #2** 14](#_Toc39425708)

[**3.2.1.3** **Use case Scenario #3** 15](#_Toc39425709)

[**3.2.1.4** **Use case Scenario #4** 16](#_Toc39425710)

[**3.2.1.5** **Use case Scenario #5** 17](#_Toc39425711)

[**3.2.1.6** **Use case Scenario #6** 18](#_Toc39425712)

[**3.2.1.7** **Use case Scenario #7** 19](#_Toc39425713)

[**3.2.1.8** **Use case Scenario #8** 20](#_Toc39425714)

[**3.2.1.9** **Use case Scenario #9** 21](#_Toc39425715)

[**3.2.1.10** **Use case Scenario #10** 22](#_Toc39425716)

[**3.2.1.11** **Use case Scenario #11** 23](#_Toc39425717)

[**3.2.1.12** **Use case Scenario #12** 24](#_Toc39425718)

[**3.2.1.13** **Use case Scenario #13** 25](#_Toc39425719)

[3.3 Class diagram 26](#_Toc39425720)

[4. External Interface Requirement 27](#_Toc39425721)

[4.1 User Interface 27](#_Toc39425722)

[4.2 Hardware Interface 27](#_Toc39425723)

[4.3 Software Interface 27](#_Toc39425724)

[4.4 Communication Interface 27](#_Toc39425725)

[5. Other Requirements 28](#_Toc39425726)

[5.1 Performance Requirements 28](#_Toc39425727)

[5.2 Safety Requirements 28](#_Toc39425728)

[5.3 Security Requirements 28](#_Toc39425729)

[5.4 Quality Software Attributes 28](#_Toc39425730)

[5.5 Usage Requirements 28](#_Toc39425731)

[Attachment A: Difficult Word List 29](#_Toc39425732)

[Flowchart 30](#_Toc39425733)

# List of Tables

[Table 2. 1 User Profile and Characteristic 8](#_Toc38753909)

[Tabel 3. 1 Functional Requirement 10](#_Toc39425668)

[Tabel 3. 2 Non-Functional Requirement 11](#_Toc39425669)

[Tabel 3. 3 Use Case Diagram 12](file:///D:\hasya\appl\SKPL_IF-42-INT_Foodordering.docx#_Toc39425670)

[Tabel 3. 4 Use Case Scenario 1 13](#_Toc39425671)

[Tabel 3. 5 Use Case Scenario 2 14](#_Toc39425672)

[Tabel 3. 6 Use Case Scenario 3 15](#_Toc39425673)

[Tabel 3. 7 Use Case Scenario 4 16](#_Toc39425674)

[Tabel 3. 8 Use Case Scenario 5 17](#_Toc39425675)

[Tabel 3. 9 Use Case Scenario 6 18](#_Toc39425676)

[Tabel 3. 10 Use Case Scenario 7 19](#_Toc39425677)

[Tabel 3. 11 Use Case Scenario 8 20](#_Toc39425678)

[Tabel 3. 12 Use Case Scenario 9 21](#_Toc39425679)

[Tabel 3. 13 Use Case Scenario 10 22](#_Toc39425680)

[Tabel 3. 14 Use Case Scenario 11 23](#_Toc39425681)

[Tabel 3. 15 Use Case Scenario 12 24](#_Toc39425682)

[Tabel 3. 16 Use Case Diagram 13 25](#_Toc39425683)

# List of Figures

[Figure 3. 1 Use Case Diagram **Error! Bookmark not defined.**](file:///D:\hasya\appl\SKPL%20Raka%20adhi%20bagaskara_19.03.2020(1).docx#_Toc38754142)

[Figure 3. 2 Class Diagram 26](#_Toc38754143)

[Figure 4. 1 Flowchart 30](file:///D:\hasya\appl\SKPL%20Raka%20adhi%20bagaskara_19.03.2020(1).docx#_Toc38754079)

# Preliminary

## 1.1 Purpose of Writing Document

Based on our experience a lot of small – scale restaurant with limited staff distress if there is a lot of customers. In this project, we create an application to help the customer order the food from the restaurant that has limited staff such as warteg to help them minimize an order error occurred.

## Document Coverage

In this SRS documentation, this document is written in Times New Roman font with the size of 12 points, using align text Justify, line spacing 1.15. Furthermore, to make it easier to read and understand the contents of this documentation, several forms of letters will be used to mark the following:

* The text written in bold is a part that must be considered.
* The text written in italics is writing that tends to be a term.

## Definition, Abbreviation, and Acronym

This application is for helping the small-scale restaurant to full-field the order for customers, the target that used this application the person who has a small-scales restaurant that every lunch or dinner has a lot of customers and sometimes has an order error occurred.

The following is a list of Definition, Abbreviation, and Acronym that will be used in the development document of Food Ordering Mobile Application for a small-scale restaurant:

1. Database: A collection of data stored systematically on a computer that can be processed or manipulated using software (application programs) to produce information.
2. Use case Diagram: Description of the process that involves an actor.
3. Class Diagram: Diagram that shows the system’s structure from class definition to build a system

## References

The references that are used to support the creation of this document is :

* Template Document that is Given from Zarina
* SKPL document by Group Naimullah (Last year)
* SKPL document from ka Sakinah

# Software Global Description

## Software Statement of Objective

A lot small-scale restaurant that has a problem with error order and limited staff that they have, based on this problem we create the application that can help them to serve the customer to the limit and to minimize the time. In our system, we create 2 user category, admin and customer. In our application, we didn’t use log in because of the random data customer that can come every day so we decided to change it with the name customer and the table where the customer sits. The admin can check the stock of the menu, notify the customer of their order status, update the stock and print the receipt. This application made by our own ideas because in Indonesia we didn’t find some small-scale restaurants that use this application. The programming language that we used is Dart language, it seems like Javascript but easier.

## Software Perspective and Function

The following are some of the product functions of the application:

1. Make ease for the customer to order foods
2. Increase efficiency of the restaurant especially if the staff number is limited

## User Profile and Characteristic

|  |  |
| --- | --- |
| **User Category** | **Task** |
| Admin | 1. Check stock of the menu 2. Notify the customer of their order status 3. Update the availability of the stock 4. Print the receipt |
| Customer | 1. Check the menu 2. Input name and table number 3. Make confirmation of the orders |

Table 2. 1 User Profile and Characteristic

## Operating Environment

These are the following of minimum hardware specifications needed to be able to run the application of food order software:

* Hardware includes:
* Android devices with a minimum of Android 5

## Software/system Boundaries

The boundary and implementation that used to develop this software such as:

* The order must be recorded on the server’s database
* The app must be able to display the form properly
* Customer must be informed if there is an invalid input
* The app must be easy to use, the menu form must be on the top, and the table number form, customer name form, and confirmation button must be on the bottom of the app
* The system should be able to record the order on the server
* The app should display a notification if the network is offline
* The app must not save the invalid form (like invalid table number or empty order)

## Assumption and Dependency

This app will be run at the most efficient if the system has:

* Android Device with a minimum of Android 7
* Have a stable internet network.
* Have a computer to act as the server with an adequate storage device

# Software Detailed Description

## Requirement Description

### Functional Requirement

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Requirement Code | Function | Description |
| 1. | FR-01 | Input order | This function enables the customer to be able to make input on the order form such as the choice of the menu, name of the customer, and the table number |
| 2. | FR-02 | Modify order | This function enables customer to customer to be able to make changes on the order in the confirmation page |
| 3. | FR-03 | Confirm order | This function enables the customer to submit the order via the submit button |
| 4. | FR-04 | Notify stock | This function enables Admin to notify the customer if the stock is unavailable |
| 5. | FR-05 | Admin confirmation | This function enables Admin to notify the confirmation of the order |
| 6. | FR-06 | Admin check stock | This function enables Admin to check the menu stock |
| 7. | FR-07 | Admin Update stock | This function enables Admin to update the stock |
| 8. | FR-08 | Print receipt | This function enables Admin to print a receipt of the orders |

Tabel 3. 1 Functional Requirement

### Non-Functional Requirement

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Quality | Requirement code | Description |
| 1. | Product Requirement | NFR-01 | This application can be used when the restaurant opens, so when the restaurant closes the application can’t be open by the customer but for admin, admin can open the application 24hr. |
| 2 | Organizational Requirement | NFR-02 | The customer just inputs the name and number of tables. |

Tabel 3. 2 Non-Functional Requirement

## Analytical Modelling

### Usecase Diagram

Tabel 3. 3 Use Case Diagram

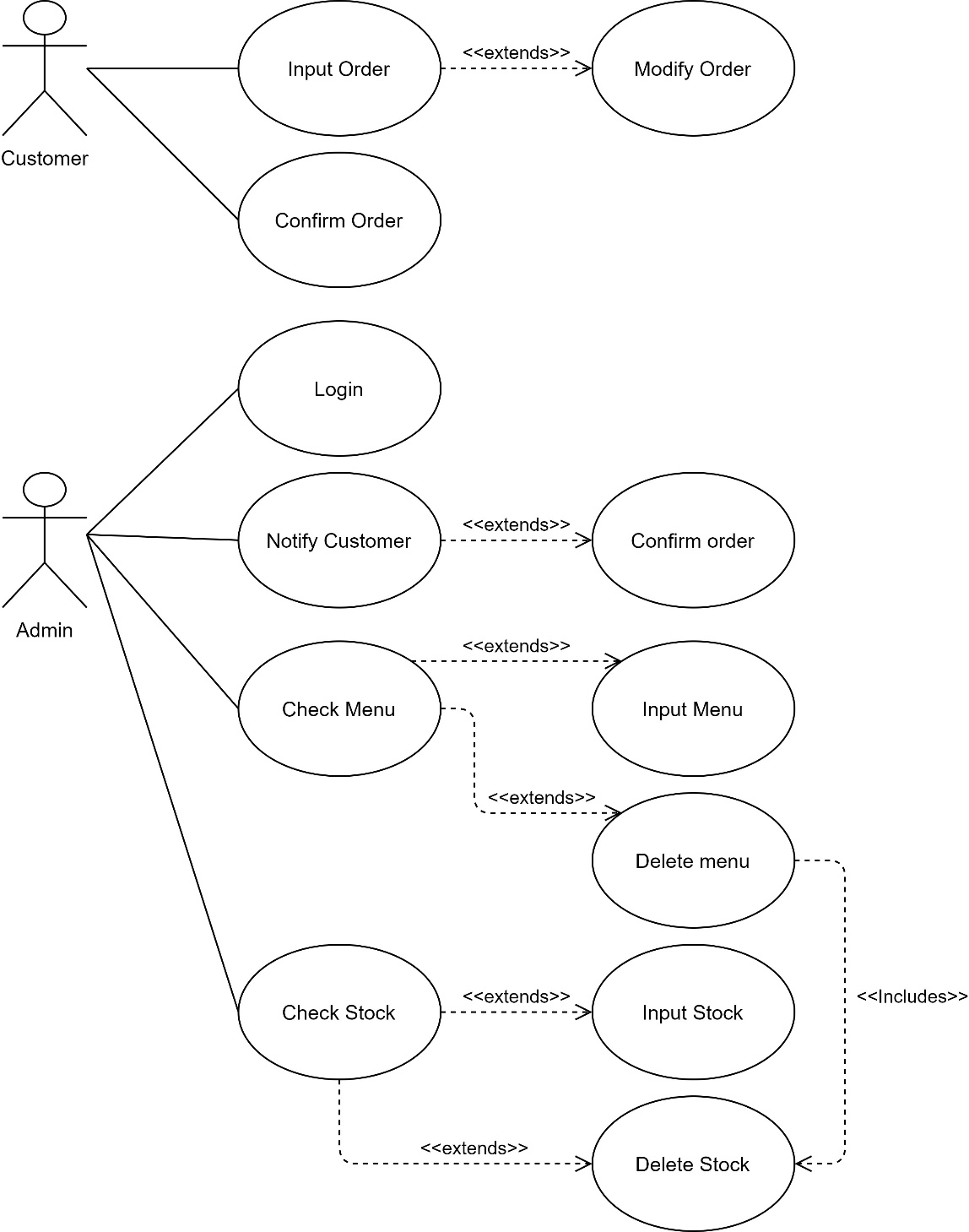


Figure 3. Use Case Diagram

#### **Use case Scenario #1**

|  |  |  |
| --- | --- | --- |
| Use Case | Input order | |
| Description | This use case is used by the customer to input their order | |
| Actor | Customer | |
| Preconditions | Customer order by tablet | |
| Postconditions | Customer have the order | |
| Main Scenario |  | |
|  | Actor | System |
| 1. Open the app |  |
|  | 2. The system displays the menu and order form |
| 3. Select the menu |  |
| 1. input the name and table number |  |
| 1. Click confirm |  |
|  | 1. Save the order in the buffer |
| Exceptional Scenario (Alternative flow) |  | |
|  | Actor | System |
| 1. Input an invalid order |  |
|  | 2. Display error message |

Tabel 3. 4 Use Case Scenario 1

#### **Use case Scenario #2**

|  |  |  |
| --- | --- | --- |
| Use Case | Modify order | |
| Description | This function is used by users to modify their orders | |
| Actor | Customer | |
| Preconditions | The customer already made their order | |
| Postconditions | Customer can change their order | |
| Main Scenario |  | |
|  | Actor | System |
|  | 1. Display confirmation page |
| 2. Customer check their order |  |
| 3. Customer click the modify order button |  |
|  | 1. System redirects to the menu page |
| Exceptional Scenario  (Alternative flow) |  | |
|  | Actor | System |
|  |  |
|  |  |

Tabel 3. Use Case Scenario 2

#### **Use case Scenario #3**

|  |  |  |
| --- | --- | --- |
| Use Case | Confirm | |
| Description | This function is used by users to confirm their orders | |
| Actor | Customer | |
| Preconditions | The customer already made their order | |
| Postconditions | Customer can confirm the orders | |
| Main Scenario |  | |
|  | Actor | System |
|  | 1. Display confirmation page |
| 2. Customer check their order |  |
| 3. Customer click the confirmation button |  |
|  | 1. The system displays the confirmation message |
| Exceptional Scenario  (Alternative flow) |  | |
|  | Actor | System |
| 1. Customer clicks the change order button |  |
|  | 2. The system redirects back to the ordering page |

Tabel 3. 6 Use Case Scenario 3

#### **Use case Scenario #4**

|  |  |  |
| --- | --- | --- |
| Use Case | Print Receipt | |
| Description | This use case is used to explain how the actor print the receipt | |
| Actor | Admin | |
| Preconditions | There is an order from the customer | |
| Postconditions | Receipt printed | |
| Main Scenario |  | |
|  | Actor | System |
| 1. Admin check for confirmed order |  |
|  | 2. The system displays the orders |
| 3. Actor prints the receipt |  |
| Exceptional Scenario (Alternative flow) |  | |
|  | Actor | System |
| 1. |  |
|  | 2. |

Tabel 3. 7 Use Case Scenario 4

#### **Use case Scenario #5**

|  |  |  |
| --- | --- | --- |
| Use Case | Notify Customer | |
| Description | This use case is to explain about the process of notifying the customer | |
| Actor | Admin | |
| Precondition | There is an order and the stock unavailable | |
| Postcondition | The customer is notified | |
| Main Scenario |  | |
|  | Actor | System |
| 1. Actor receives new order |  |
|  | 2. System display the orders |
| 3. The actor check the availability of the stock |  |
| 4. Actor send message to customer |  |
|  |  | 5 System deliver the message to customer |
| Exceptional Scenario  (Alternative flow) |  | |
|  | Actor | System |
| 1. |  |
|  | 2 |
|  |  |  |

Tabel 3. 8 Use Case Scenario 5

#### **Use case Scenario #6**

|  |  |  |
| --- | --- | --- |
| Use Case | Confirm order | |
| Description | This function is used by Admin to confirm the order | |
| Actor | Admin | |
| Preconditions | There is a new order | |
| Postconditions | The order is confirmed | |
| Main Scenario |  | |
|  | Actor | System |
|  | 1. Display order information |
| 2. Admin check the order |  |
| 3. Admin click confirm button |  |
|  | 1. System save the order confirmation and notify the customer that their order has been confirmed |
| Exceptional Scenario  (Alternative flow) |  | |
|  | Actor | System |
|  |  |
|  |  |

Tabel 3. 9 Use Case Scenario 6

#### **Use case Scenario #7**

|  |  |  |
| --- | --- | --- |
| Use Case | Check Menu | |
| Description | This use case is used to explain how the actor check the menu | |
| Actor | Admin | |
| Precondition | A menu data is present on the database | |
| Postcondition | Menu has been edited | |
| Main Scenario |  | |
|  | Actor | System |
| 1.Open the menu description page |  |
|  | 2. The system display the menu description page |
| 3. Actor checks for the menu that available on the database and the restaurant |  |
| 4. Actor click “input” button |  |
|  | 5. System will add the input to menu database |
| Exceptional Scenario (Alternative flow) |  | |
|  | Actor | System |
|  |  |
|  |  |

Tabel 3. 10 Use Case Scenario 7

#### **Use case Scenario #8**

|  |  |  |
| --- | --- | --- |
| Use Case | Input menu | |
| Description | This function is used by admin to input a menu | |
| Actor | Admin | |
| Preconditions | Admin is on the check menu page | |
| Postconditions | Admin can input a menu | |
| Main Scenario |  | |
|  | Actor | System |
|  | 1. Display input menu form |
| 2. Admin input menu details |  |
| 3. Admin click add button |  |
|  | 1. System added the menu on the database |
| Exceptional Scenario  (Alternative flow) |  | |
|  | Actor | System |
|  |  |
|  |  |

Tabel 3. 11 Use Case Scenario 8

#### **Use case Scenario #9**

|  |  |  |
| --- | --- | --- |
| Use Case | Delete menu | |
| Description | This function is used by admin to delete the menu | |
| Actor | Admin | |
| Preconditions | Admin is in the check menu page | |
| Postconditions | Admin deleted a menu | |
| Main Scenario |  | |
|  | Actor | System |
|  | 1. Display menu details page |
| 2. Admin select the menu |  |
| 3. Admin click delete button |  |
|  | 1. System deletes the menu and the menu’s stock as well |
| Exceptional Scenario  (Alternative flow) |  | |
|  | Actor | System |
|  |  |
|  |  |

Tabel 3. 12 Use Case Scenario 9

#### **Use case Scenario #10**

|  |  |  |
| --- | --- | --- |
| Use Case | Check Stock | |
| Description | This function is used by admin to check the availability of the stock | |
| Actor | Admin | |
| Preconditions | There is stock present on the database | |
| Postconditions | The stock is edited | |
| Main Scenario |  | |
|  | Actor | System |
| 1. Actor open the stock details page |  |
|  | 2. System displays the stock details page |
| 3. Actor input the stock for the item |  |
|  | 4. System added the stock of the item |
| Exceptional Scenario  (Alternative flow) |  | |
|  | Actor | System |
|  |  |
|  |  |

Tabel 3. 13 Use Case Scenario 10

#### **Use case Scenario #11**

|  |  |  |
| --- | --- | --- |
| Use Case | Input stock | |
| Description | This function is used by the admin to input the stock of the menu | |
| Actor | Admin | |
| Preconditions | Admin is in the check stock page | |
| Postconditions | Admin added the stock | |
| Main Scenario |  | |
|  | Actor | System |
|  | 1. Display stock details page |
| 2. Admin select the menu |  |
| 3. Admin click add button at the respective menu |  |
|  | 1. System added the stock of the menu |
| Exceptional Scenario  (Alternative flow) |  | |
|  | Actor | System |
|  |  |
|  |  |

Tabel 3. 14 Use Case Scenario 11

#### **Use case Scenario #12**

|  |  |  |
| --- | --- | --- |
| Use Case | Delete Stock | |
| Description | This function is used by the admin to delete the stock of the menu | |
| Actor | Admin | |
| Preconditions | Admin is in the check stock page | |
| Postconditions | Admin deleted the stock | |
| Main Scenario |  | |
|  | Actor | System |
|  | 1. Display stock details page |
| 2. Admin select the menu |  |
| 3. Admin click delete button at the respective menu |  |
|  | 1. System deleted the stock of the menu |
| Exceptional Scenario  (Alternative flow) |  | |
|  | Actor | System |
|  |  |
|  |  |

Tabel 3. 15 Use Case Scenario 12

#### **Use case Scenario #13**

|  |  |  |
| --- | --- | --- |
| Use Case | Login | |
| Description | This function is used by the admin to login to the menu | |
| Actor | Admin | |
| Preconditions | Admin can’t log in | |
| Postconditions | Admin can log in | |
| Main Scenario |  | |
|  | Actor | System |
|  | 1. Display the login page |
| 2. Admin input username and password |  |
|  | 3. System display the admin page |
| 3. Admin can log in |  |
| Exceptional Scenario  (Alternative flow) |  | |
|  | Actor | System |
|  |  |
|  |  |

Tabel 3. 16 Use Case Diagram 13

## Class diagram

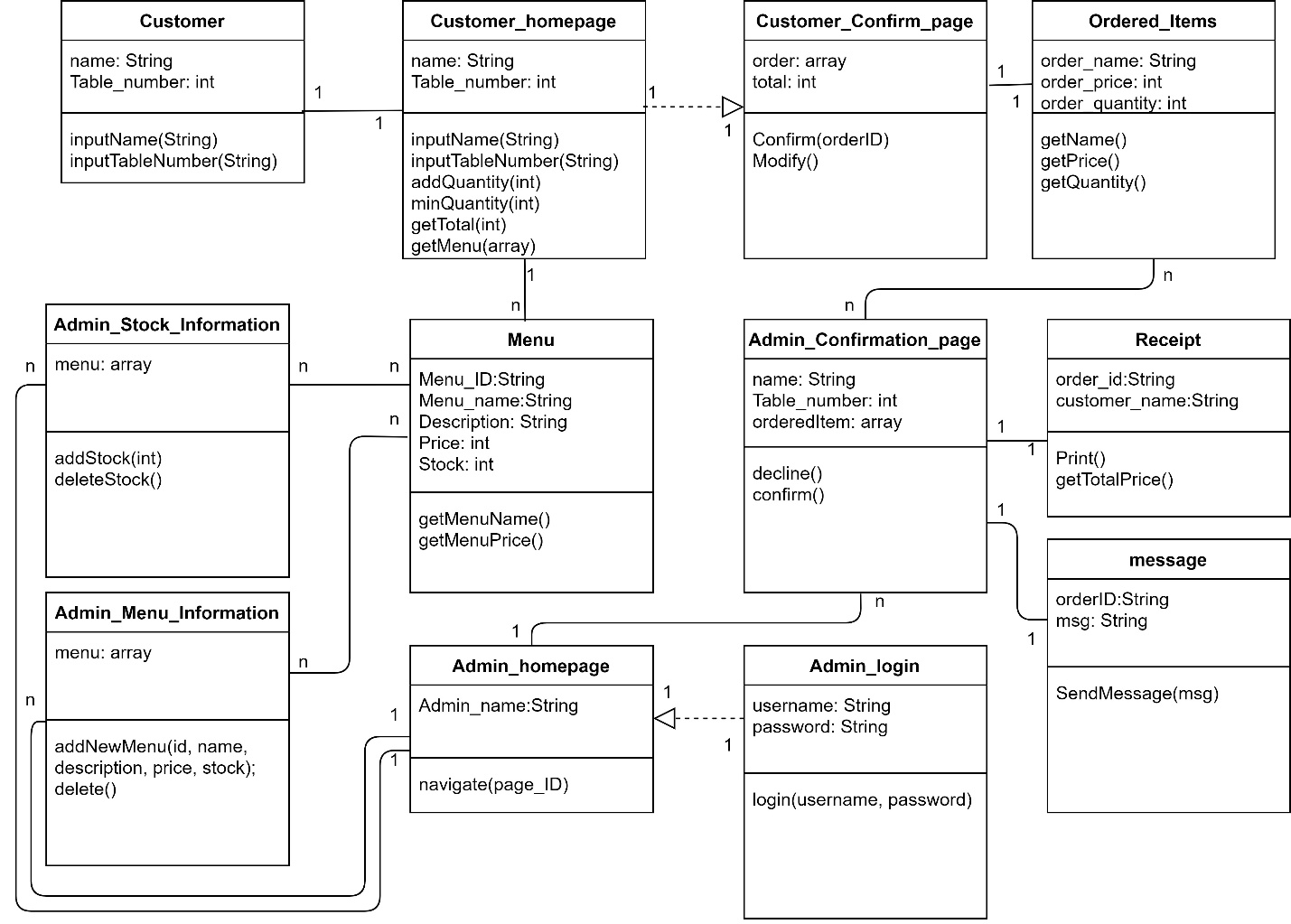


Figure 3. 2 Class Diagram

# External Interface Requirement

## 4.1 User Interface

Users will be able to access the apps without log in to data base, so customer will see the menu and price of the food. for admin it will be able for edit the menu and price, delete the menu if there is a shortage of food or food ingredients, there isn't any and input the menu if there is a new menu available.

## 4.2 Hardware Interface

In this case, we use tablet gadget as our platform that required internet connection. Here are the Minimum Specification Requirement for the tablet gadget :

* Memory 512 MB RAM
* 1.0 GHz Dual Core Processor
* 720p Display Resolution
* Wi-Fi

So, if the customer want to order something, they need to use the tablet as the order platform with the Minimum System Requirement above.

## 4.3 Software Interface

Food Ordering Mobile Application for Small-Scale Restaurant is a program that will be built with Dart Language and will run with operating system Android.

## 4.4 Communication Interface

The application that we make is for customers that to facilitate customers ordering food through this application. So in this application customer can order food base on application and can make additional the order after customer complete the order and confirm the order. After the order system display the customer order admin check the order from customers admin will check the available of the stock and actor will send the notification the customers and the system will deliver the messages to the customer and admin will prepare for the order customer. And for admin can check the menu description page, the system will display the description page after that actor check the menu that available on the database and the restaurant, admin can input button and system will add the input to menu database after that admin can open the stock detail page, the system will display the stock page, actor can input the stock for the item and system will added the stock of the item. Admin can delete the menu if the stock are not available.

# Other Requirements

## 5.1 Performance Requirements

The system that we develop takes less than 1 minute to access every page on it. To search the data, this system needed time to response less than 2 minutes.

## 5.2 Safety Requirements

We built this safety system with the system that is suitable to the users requirements. Every admin and the workers has the responsibility also the policy to secure the system to make it functionate and facilitate when the user access it.

## 5.3 Security Requirements

In this application we didn’t use the log in for customer and admin.

## 5.4 Quality Software Attributes

These are the boundary of that system:

* This program developed using Dart Language.
* This system is developed using MySQL database.
* The server that used in this system is using Windows.

## 5.5 Usage Requirements

These are the application usability:

* User interface for this application using CSS and Dart language.
* Language used in this system is English

# Attachment A: Difficult Word List

· Actor: A human entity/machine that interacts with system to do some works.

· Hardware: Tools, machinery, and other durable equipment.

· Software: The programs and other operating information used by a computer.

· SKPL: Spesifikasi Kebutuhan Perangkat Lunak, document analysis result that contains software specifications.

· Use Case: Description of process that involves actor.

· Class Diagram: Diagram that shows the system’s structure from classes definition to build a system.

· User: The person who’s using the system

· Admin: Person that has more access to control the system’s content.

# Flowchart

Figure 4. Flowchart