**SYSTEM REQUIREMENTS SPECIFICATION DOCUMENT**

Document Number: BBT 3104-SyRS-2020/04

Version: 3.7

Review Date:

Submission Date: 7/14/2020

Case Study: Wanji’s Food Industries

Group Number: 36

Team Members:

|  |  |  |
| --- | --- | --- |
|  | **Student Details** | **Mark Awarded** |
|  | Lilian Thairu – 111878. – Group A – SU Email: Lilian.Thairu@strathmore.edu |  |
|  | Lydia Kahuria – 110875. – Group A– SU Email:Lydia.Kahuria@strathmore.edu |  |
|  | Fredah Kioko– 103428. – Group – SU Email:Fredah.Kioko@strathmore.edu |  |

|  |  |  |
| --- | --- | --- |
| **Version Change** | **Date of Change** | **Changes Made** |
| Version 1.0 to 1.2 | 25th April 2020 | Added the business process and objectives |
| Version 1.2 to 1.6 | 9th May 2020 | Added the system capabilities, Added the system constraints, Added the conditions |
| Version 1.6 to 2.1 | 22nd May 2020 | Added ERD and the Uml, database Schema and the system Architecture |
| Version 2.1 to 2.7 | 19th June 2020 | Added the Dependencies |
| Version 2.7 to 2.9 | 26th June 2020 | Added 1st Normal form and he 2nd Normal form |
| Version 2.9 to 3.1 | 3rd July 2020 | Added 3rd ,BOYCE and 5th Normal form |
| Version 3.1 to 3.6 | 10th July 2020 | Added The search engine , the procedures,functions,views |
| Version 3.6 to 3.7 | 12th July 2020 | Added all the DDL statements,updated the list of figures,the list of tables and wrote the abstract. |

# Executive Summary

***<This will be filled in at the end. It is like an abstract>***

Wanji’s food production is a new

**Table of Contents**

[Executive Summary 1](#_gjdgxs)

[List of Figures 3](#_30j0zll)

[List of Tables 4](#_1fob9te)

[List of Models 5](#_49x2ik5)

[Chapter 1.](#_3znysh7) Introduction 6

[1.1](#_2et92p0) The Mission of the Business 6

[1.2](#_tyjcwt) The Day to Day Operations of the Business 6

[1.3](#_3dy6vkm) Statement of Objectives: Aim of Designing and Implementing the Database 6

[Chapter 2.](#_1t3h5sf) System Capabilities 7

[A.](#_4d34og8) Example1 Data (E.g. Patient Data or Client Data or Bicycle Data) 7

[B.](#_2s8eyo1) Example2 Data 7

[C.](#_17dp8vu) Example3 Data 7

[Chapter 3.](#_3rdcrjn) System Conditions 8

[A.](#_26in1rg) Example1 Data (E.g. Patient Data or Client Data or Bicycle Data) 8

[B.](#_35nkun2) Example2 Data 8

[C.](#_1ksv4uv) Example3 Data 8

[Chapter 4.](#_44sinio) System Constraints 9

[A.](#_2jxsxqh) Example1 Data (E.g. Patient Data or Client Data or Bicycle Data) 9

[B.](#_z337ya) Example2 Data 9

[C.](#_3j2qqm3) Example3 Data 9

[Chapter 5.](#_1y810tw) Assumptions 10

[Chapter 6.](#_2xcytpi) Database Design 11

[a.](#_1ci93xb) Entity Relationship Diagram Based on the Chen Notation 11

[b.](#_3whwml4) Database Schema Based on the UML Notation 12

[Chapter 7.](#_2bn6wsx) Normalization 13

[a.](#_qsh70q) List of Functional Dependencies 13

[b.](#_3as4poj) Normal Forms 13

[Chapter 8.](#_1pxezwc)

Chapter 9.

# List of Figures

# List of Tables

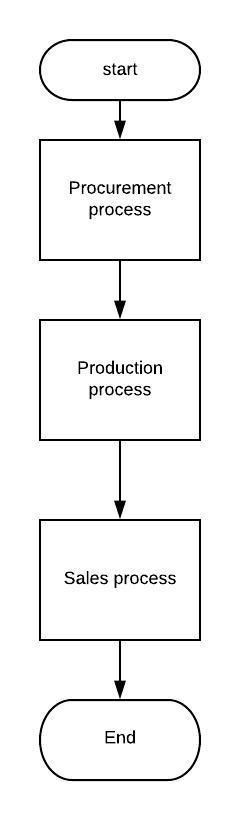
# Introduction

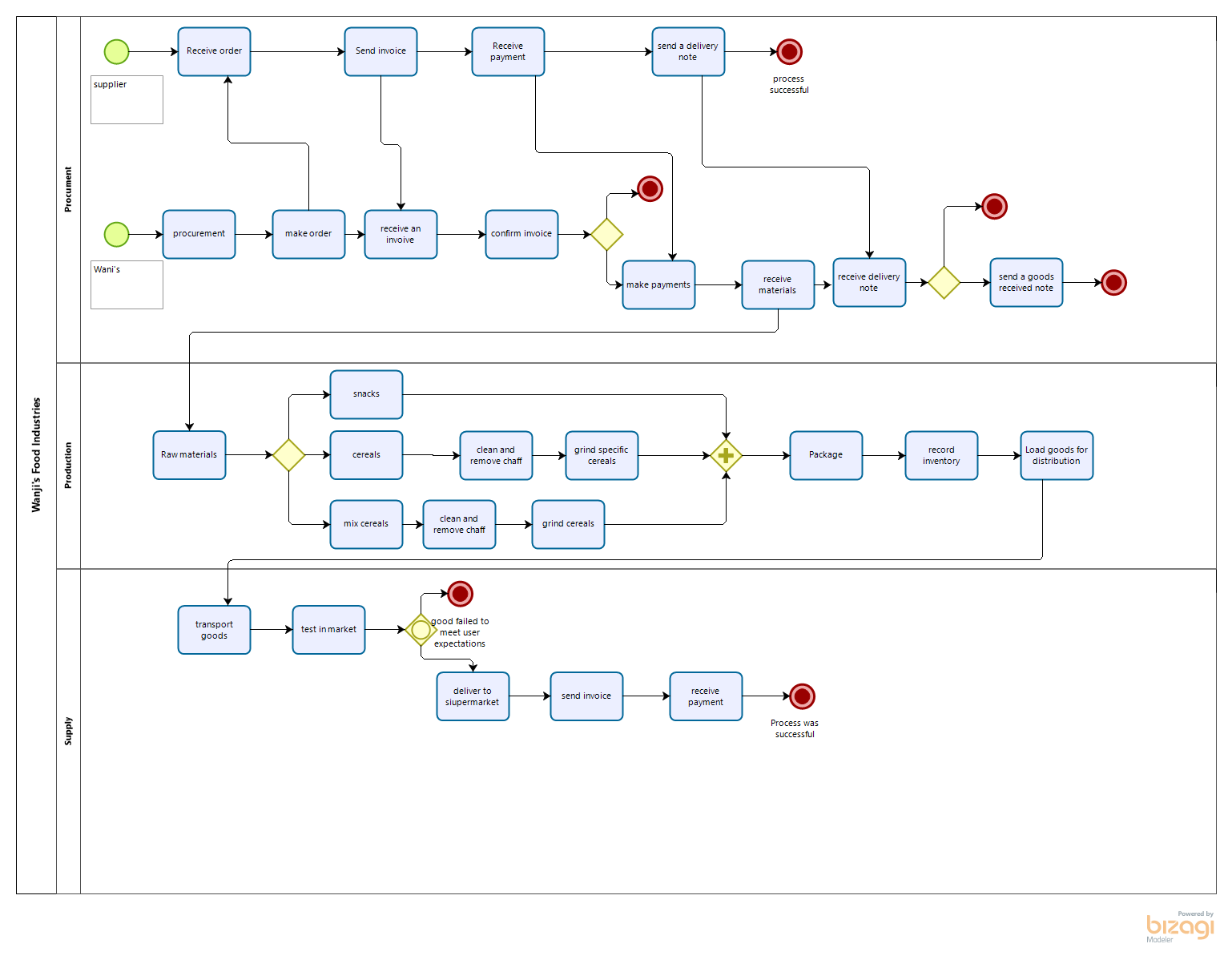
## The Mission of the Business

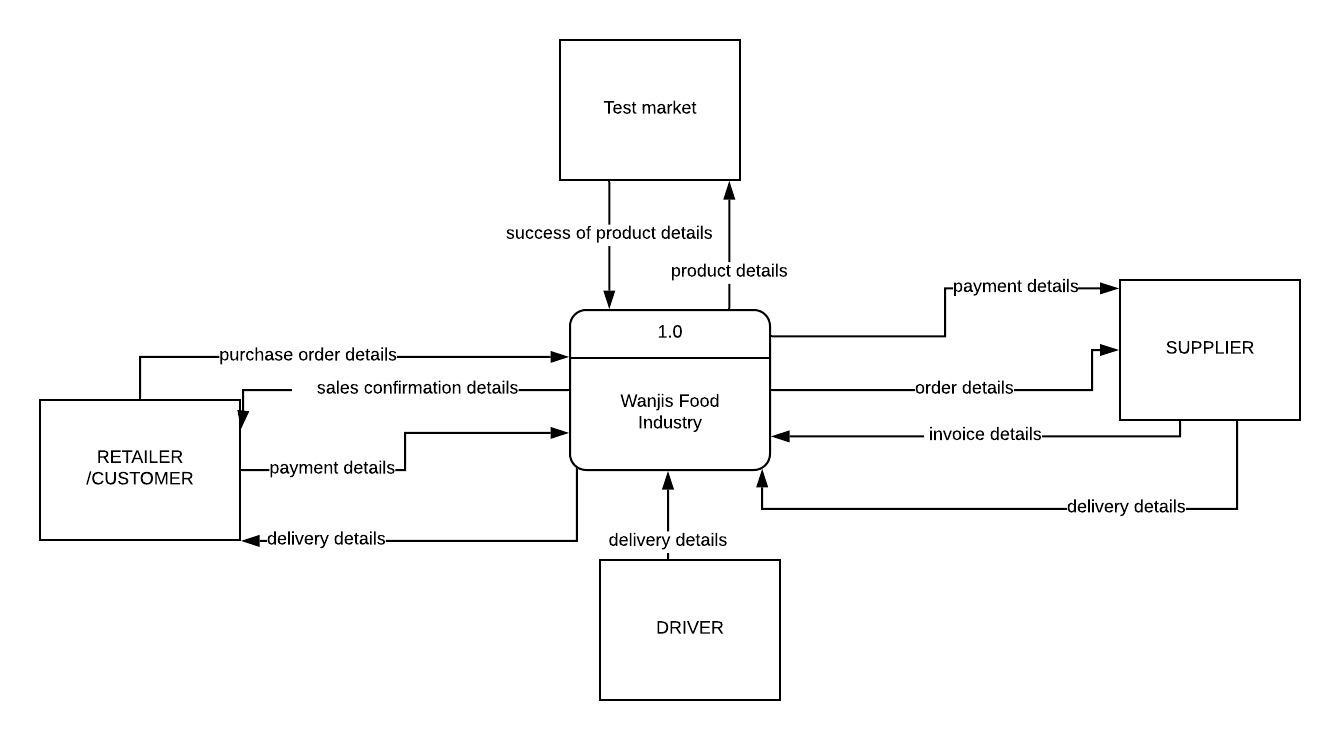
The mission of Wanji’s Food Industries is to process healthy and nutritious food products targeted to a specific group of people with specific needs hence solving the societal problem of malnutrition by making them affordable to the consumers.

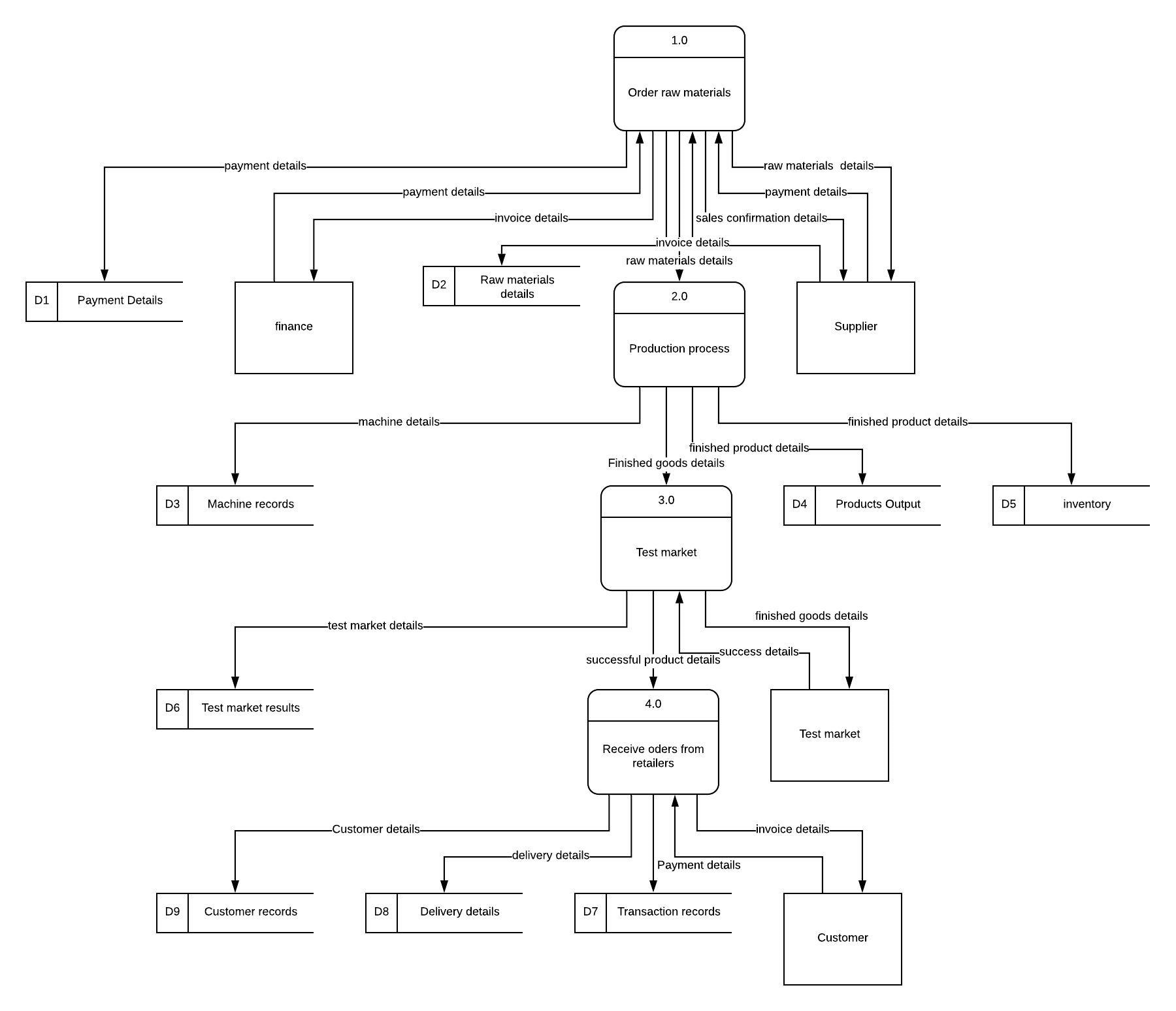
## The Day to Day Operations of the Business

***<Create a flowchart here that represents the business processes involved in producing the business' main product/service>***





***<Create a Data Flow Diagram (DFD) that represents the flow of information through the business organization in the case study>***



## Statement of Objectives

***<Hint: Think in terms of an IT-based solution to address the negative effects associated with the non-value adding activities in the organization’s business processes>***

1. To record the inventory of the organization in a database in order to generate reports that may be used to make decisions in the future.
2. To store customer orders in order to know the demand of the products and the success rate of each product in a database management system
3. To analyze the stored Purchase details of raw materials against the output of products processes by the industry using business intelligence dashboard
4. To manage staff details and know the activeness of each member and any bonuses if any by creating relationships and automating the calculations of salaries in the database
5. To compare sales of different products to know the performance of the food processing industry using a graph and displayed in the business intelligence dashboard
6. To track payments made to the supplier and those received from retailers
7. To track the quality of raw materials given by the supplier by recording the supplier details.

# System Capabilities

## Employee Details Data

***<Guiding example: The database should be capable of storing data about the clients and for each client, the following data should be stored:>***

|  |  |
| --- | --- |
| **1.** | Employee\_ ID |
| **2.** | Employee\_ First name |
| **3.** | Employee \_Last Name |
| **4.** | Employee\_Contacts |
| **5.** | Employee\_ Department |
| **6.** | Employee\_Salary |
| **7.** | Employee\_Bonus/Allowances |
| **8.** | Status |

## Raw Materials Data

## *<Guiding example: The database should be capable of storing data about the rental properties and for each rental property, the following data should be stored:>*

|  |  |
| --- | --- |
| **1.** | Raw Material ID |
| **2.** | Raw Material Name |
| **3.** | Product Output |

## Supplier Payment Details(SP)

|  |  |
| --- | --- |
| **1.** | SP Transaction Number |
| **2.** | SP Method of payment |
| **3.** | SP Invoice received |
| **4.** | SP Name of supplier |
| **5.** | SP Amount paid |
| **6.** | SP Day of payment |

## D. Retailer Payment Details Data

|  |  |
| --- | --- |
| **1.** | Retailer Transaction Number |
| **2.** | Retailer Method of payment |
| **3.** | Retailer Name of supplier |
| **4.** | SP Retailer Amount paid |
| **5.** | Retailer Day of payment |

## E. Supplier Details Data

|  |
| --- |
| 1. Supplier Id |
| 2. Supplier Company Name |
| 1. Supplier Invoice |

## Sales Details Data

|  |  |
| --- | --- |
| **1.** | Sales Product ID |
| **2.** | Sales Product Name |
| **3.** | Number of products sold |
| **4.** | Date of Sales |

## Product Details Data

|  |  |
| --- | --- |
| **1.** | Product ID |
| **2.** | Product Name |
| **3.** | Manufacture\_Date |
| **4.** | Raw\_Materials ID |

## Machine Details Data

|  |  |
| --- | --- |
| **1.** | Machine Number |
| **2.** | Machine Type |
| **3.** | Machine Active Years |

## 4. Machine Output

## I. Order Details Data

|  |  |
| --- | --- |
| **1.** | Order Id |
| **2.** | Order details |
| **3.** | Order Invoice |
| **4.** | Delivery Note |
| **5.** | Date of order |

## I. Test Market Details Data

|  |  |
| --- | --- |
| **1.** | Test Market ID |
| **2.** | Test Market location |
| **3.** | Test Retailer Name |
| **4.** | Test product Name |
|  | Test Duration |
| **6.** | Test Success status |

# System Conditions

## Employee Details Data *<Client Data>*

***<Guiding example: The database should contain the following measurable characteristics for each capability specified in Chapter 2.>***

|  |  |
| --- | --- |
| **1.** | The employee Identification is their National ID. Each National ID is unique to a person. |
| **2.** | The employees should fill in their first name which is either one or two. |
| **3.** | The employees should fill out their last name as well which is either one or two as well. |
| **4.** | Telephone numbers will be required so that the employee can be contacted easily in case of an unexpected outcome. So employees may have more than one phone number but only two most reachable contacts will be necessary. |
| **5.** | Every employee should fill in the department they are featured in, that is, where they mainly do their work for example the production department as well as the accounts department. One employee belongs to one department. |
| **6.** | The amount of salary received is also information required in the database. The salary can be written in thousands or even millions, but it can only be once. |
| **7.** | When employees receive bonuses, in essence, an extra amount of cash from the normal salary, it needs to be recorded in the database. This can be written in thousands only, and can only be filled in once. |
| **8.** | Status implies to the position one holds in the company hence, this is information crucial to the company that must be recorded in the database. One employee holds one position. |

## Raw Material Data

|  |  |
| --- | --- |
| **1.** | The product ID is the number that is assigned to the product (Serial number) as received from the supplier.Here it will act as the raw materials needed in production This number is unique to each product. This is also the number used in scanning for sales. |
| **2.** | The raw materials can have products of one or more names. |
| **3.** | This is the number of hours and the amount of that were used in the production of the product, it can be measured using the number of hours. |

## Supplier Payment Details Data

|  |  |
| --- | --- |
| **1.** | The transaction number is unique to a specific supplier. (This is the foreign key for the supplier table) |
| **2.** | Many suppliers can be paid using the same method of payment. |
| **3.** | A supplier delivers their own invoice hence 1:1 |
| **4.** | Two or more suppliers may share the same name hence, m:1 |
| **5.** | One payment is specific to the supplier who is being paid. |
| **6.** | Many suppliers can pay on the same day, hence m:1. |

# Retailer Payment Details Data

|  |  |
| --- | --- |
| **1.** | The transaction number is unique to a specific retailer. |
| **2.** | Many retailers can pay using the same method of payment. |
| **3.** | A retailer is given their own invoice hence 1:1 |
| **4.** | Two or more retailers may share the same name hence, m:1 |
| **5.** | Many retailers can pay on the same day, hence m:1 |

# Order details Data

|  |  |
| --- | --- |
| **1.** | The Company’s ID is unique to each supplier, hence which includes the Companies identification number, 1:1 |
| **2.** | Order details may share a common list of requirements, m:1 |
| **3.** | Order Invoice Each supplier delivers their own unique type of invoice, 1:1 |
| **4.** | Oder Date Wanjii’s Food industries may opt to send orders on the same day |
| **5.** | Many goods may come from the same supplier m:1 |

# Sales Details Data

|  |  |
| --- | --- |
| **1.** | The product ID is unique to every product. |
| **2.** | A product may have a common name, hence, 1:1 |
| **3.** | The number of a product that has been sold may be the same as another product, hence m:1 |

# Inventory Details Data

|  |  |
| --- | --- |
| **1.** | The item number is specific to an item, 1:1 |
| **2.** | Many goods from different suppliers may be received on the same day, hence, m:1 |
| **3.** | The quantity of the number of orders sent may be the same for different products hence, m:1 |

# Machine Details Data

|  |  |
| --- | --- |
| **1.** | The machine number is specific to a machine,in this case it is the build number of the machine 1:1 |
| **2.** | Each machine performs a different functionality, hence, 1:1 |
| **3.** | The years active for machines may be the same hence, m:1 |

# 4. The machine output is the amount of products the machine can process in this the output

# depends on the amount of raw materials used and the amount of minimise wastages hence m:1

## I Order Details Data

|  |  |
| --- | --- |
| **1.** | Order ID is the number provided by the procurement document number It should be unique to every order |
| **2.** | Order Details should be the list of raw materials to the supplier |
| **3.** | Order Invoice this is the invoice received from the supplier |
| **4.** | Delivery Note this is the document with the list of goods delivered to the organization from the supplier |
| **5.** | Date of order this is the date that the raw materials were orderd from the supplier |

# System Constraints

***<E.g. interrelational, intarelational, static, dynamic, semantic, primary key, & foreign key>***

*E.g. The database must satisfy the following constraints:*

## Employee Details Data

|  |  |
| --- | --- |
| **1.** | The Employee Identification number, first name and Last name shall have a key constraint. It will be intrarelational, composite, static key constraint to form the primary key. The Employee ID domain constraint is VARCHAR |
| **2.** | First name shall have a key constraint,that is dynamic and is intrarelational with the last name and Employee ID. The domain constraint is STRING |
| **3.** | The last name shall have a dynamic key constraint as well and is intrarelational with the first name and Employee ID. The domain constraint is STRING |
| **4.** | Contacts shall have a static constraint that is intrarelational with the First Name and Last Name. The domain constraint is BOOLEAN |
| **5.** | Department is intrarelational with the salary and bonuses but is interrelational with the status. The domain constraint is STRING |
| **6.** | Salary is intrarelational with the Department and the status but is interrelational with the bonus. It is also a semantic key constraint as the employee cannot earn more than the employer. The domain constraint is BOOLEAN |
| **7.** | The Bonus is semantic,static,intrarelational with the Department and status but interrelational with the Salary. The Domain constraint is BOOLEAN |
| **8.** | The Status is intrarelational with the salary and the bonus but interelational with the Department. The domain constraint is STRING |

## Raw Materials Data

|  |  |
| --- | --- |
| **1.** | The Raw material ID shall have a static,interrelational,composite constraint to the Product name and intrarelational to the output of the product. It is the primary key for the products data and also the Foreign key for the Sales Details Data and the Inventory Details Data. The domain constraint is BOOLEAN |
| **2.** | The Raw Material Name is dynamic,intrarelational to the output of the product and interrelational to the Product ID. The domain name is STRING |
| **3.** | The output of the product shall have a dynamic constraint that is Intrarelational to the Product ID and Product Name and is interrelational to the number of products sold. The domain constraint is BOOLEAN |

## Supplier Payment Details Data

|  |  |
| --- | --- |
| **1.** | The transaction number is interrelational,composite to the Company ID of the supplier, the Name of the supplier and the method of payment used. The transaction Number forms the Primary key for the supplier Payment Details Data. The Domain constraint is BOOLEAN |
| **2.** | The method of payment shall have a static constraint that is interelational to the transaction number, the Company ID and the Company Name. The domain constraint is BOOLEAN |
| **3.** | The invoice received is intrarelational to the amount paid and the day of payment but it is interrelational to the name of the supplier. The domain constraint is BOOLEAN |
| **4.** | The name of the supplier is intrarelational to the amount paid, the invoice received and Day of Payment. The domain constraint is STRING |
| **5.** | The amount paid is intrarelational to the name of supplier but is interrelational to the transaction number and the invoice received. The domain constraint is BOOLEAN |
| **6.** | The Day of payment is interrelational to the amount paid and the invoice received. The domain constraint is DATE |

# Retailer Payment Details Data

|  |  |
| --- | --- |
| **1.** | The transaction number will have a composite, interrelational to the Name of the retailer and the method of payment. The Domain constraint is BOOLEAN |
| **2.** | The method of payment is interrelational to the transaction number and the Retailers’ Name. The domain constraint is BOOLEAN |
| **3.** | The name of the retailer is intrarelational to the amount paid and Day of Payment. The domain constraint is STRING |
| **4.** | The amount paid is intrarelational to the name of the retailer but is interrelational to the transaction number. The domain constraint is BOOLEAN |
| **5.** | The Day of payment is interrelational to the amount paid and the Delivery note. The domain constraint is DATE. |

# Supplier Details Data

|  |  |
| --- | --- |
| **1.** | The Supplier ID is interrelational to the Transaction number for the payment as well as the Company Name. The Company ID forms the primary key for the supplier Details Data and is the Foreign key for the supplier Details Data. The domain constraint is BOOLEAN |
| **2.** | The Order Details is interrelational to the Company ID. The domain constraint is STRING |
| **3.** | The Invoice is intrarelational to the Company ID and Company Name. The Domain constraint is VARCHAR |
| **4.** | The supplier\_Company Name is the attribute key to this particular table. The Domain Constraint is STring |
| **5.** | The Date of order is intrarelational to the Company ID and the Company Name. The domain constraint is DATE |
| **6.** | The delivery note is interelational to the company ID and the raw materials. The domain constraint is VARCHAR |

## Sale Details Data

|  |  |
| --- | --- |
| **1.** | The product ID will have a composite, interrelational constraint to the Product name and intrarelational to the output of the product and number of products sold.I It will also has a referential constraint The domain constraint is BOOLEAN |
| **2.** | The product Name is intrarelational to the Output of the product, Number of Products sold and interrelational to the Product ID. The domain name is STRING |
| **3.** | The number of products sold is Intrarelational to the Product ID and Product Name and interrelational to the Output of the product. The domain constraint is BOOLEAN |

## Product Details Data

|  |  |
| --- | --- |
| **1.** | The Product ID will have a composite semantic, interrelational to the Product name and Product ID while intrarelational to the output of the product and the Number of products sold. The domain constraint is BOOLEAN |
| **2.** | The Product Name received is intrarelational to the Deliver Note, Invoice Received, Product ID, Product Name and Amount Paid. The domain name is DATE |
| **3.** | The manufacture Date is Intrarelational to the Product ID, Product Name and Amount Paid. The domain constraint is BOOLEAN |

## 

## Machine Details Data

|  |  |
| --- | --- |
| **1.** | The Machine Number is intrarrelational to the build number when the machine was purchased. The domain constraint is VARCHAR |
| **2.** | The Machine Type is intrarelational to the Machine Number The domain name is STRING |
| **3.** | The Years active have a domain constraint that is BOOLEAN |

4. The Machine output shall have an intrarelation constraint.

# Order Details Data

|  |  |
| --- | --- |
| **1.** | The Oder ID will have a compositeDomain constraint is BOOLEAN |
| **2.** | The order details is interrelational to the transaction number and the Retailers’ Name. The domain constraint is BOOLEAN |
| **3.** | The Oder invoice is intrarelational to the amount paid and Day of Payment. The domain constraint is STRING |
| **4.** | The delivery note is intrarelational to the name of the retailer but is interrelational to the transaction number. The domain constraint is BOOLEAN |
| **5.** | The Day of order Delivery note. The domain constraint is DATE. |

# Assumptions

It is assumed that wanjis food industries has a stable relationship with all its suppliers

It is assumed that all The payment transactions occur via the bank or mpesa

It is assumed that all the food processing machines are of the same brand

It is also assumed that the suppliers are paid using the same method.

It is assumed that all orders are delivered on time with minimal or no delays.

It is assumed that the machines used by Wanjis Food industry are of good quality and in good working condition.

It is assumed that employees may receive bonuses or allowances.

It is assumed that all the temporary employees come back when needed and do not change.

It is assumed that each employee working in the manufacturing stage has a specific experty hence only operates a specific machine in which they are qualified.

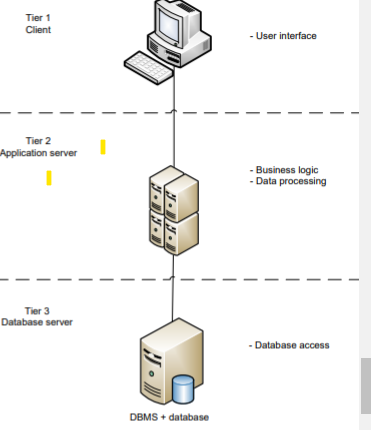
It is assumed that all Suppliers have company ID’S that are unique to them

# Database Design

## Entity Relationship Diagram Based on the Chen Notation

## Database Schema Based on the UML Notation

## System Architecture



These are the advantages of a 3 tier system architecture

1) Less expensive hardware for the client in tier 1

2) Eliminates the concerns of software distribution because the core of the application is installed on the centralized application server

3) Modularity enables one tier to be replaced/modified without affecting the other tiers

4) Load balancing is easier with the separation of the core business logic (coded as algorithms in a software: the algorithms process the data based on predefined business logic documented in SOPs, e.g. an algorithm to calculate the Net Present Value) from the database functions (storage and retrieval of data) I saw it best to use a three-tier architecture because of the advantages that comes with it. First, this type of architecture is very scalable whereby the application servers can be used in many machines and the database does not require direct connection to every client. This allows for business growth over time. The company also needs computers that are going to act as clients, each computer costs about $800 with good specifications and capability to be able to handle all the system - User interface - Main business logic and data processing logic Tier 1 C

# Normalization

## List of Functional Dependencies

***E.g. Full, Partial, Transitive, Multi-Valued Dependencies (MVD), Join Dependencies (JD), etc.***

Full Function Dependencies

The product number is fully dependent on the product Id and the product type.

The supplier delivery note is fully dependent on the supplier Id and the Supplier Company name.

The delivery note is fully dependent on the key constraints that is The order ID, the order details, the invoice of the order and the date of the order.

The Order invoice is fully dependent on the key constraints that is The order ID, the order details and the date of the order.

Retailer order is fully functionally dependent on the retailer id the retailer name and the retailer location.

The date of sales fully depends on the number of products sold, the retailer id and the product name.

Partial Dependency

The supplier transaction number partially depends on the supplier supplier ID and the date of the transaction.

The machine output partially depends on the raw material input and the Employee id.

Transitive Dependency

The employee contacts transitively depend on the Employee ID

The employee last name and employee first name transitively depends on the Employee id

The supplier company name transitively depends on the supplier Id

Multi-Valued Dependency

Machine Id is a multi- determinant of the Employee\_Id and the machine years active.

The Retailer Id multi-determines the retailer location and the retailer order

The Supplier Id multi-determines the Supplier payment of method and the date of the transaction

The Employee Id multi-determined the Employee\_Department and the employee salary and the date of registration

Join Dependency

**Employee**(Employee\_id(), Employee\_First ,Employee\_lastname )

Can be decomposed into

## Normal Forms

***i.e. 1NF, 2NF, 3NF, BCNF, 4NF, and 5NF***

***1NF***

***2NF***

***3NF***

***BCNF***

***4NF***

***5NF***

# Advanced Database Objects

## a. Storage Engines

# *List the storage engine that each relation will use*

The table will use CSV,INNODB,MERGE,ARCHIVE,FEDERATED

CREATE TABLE **Employee** (i INT) ENGINE = INNODB;

CREATE TABLE **Retailer** (i INT) ENGINE = CSV; ‘

CREATE TABLE **Test\_Market**(i INT) ENGINE = MEMORY;

CREATE TABL**Supplier**(i INT) ENGINE = MEMORY;

## b. Indexes

# *List all the attributes that will be indexed as well as the type of index that will be used*

# Primary key,super key,candidate,composite foreign…hash index-used in memory and NDB,cluster b+ tree index used in a join,r tree, ndb uses hasdh

## c. Normal Triggers

# *List all the normal triggers and how they will be used to enforce the database constraints*

Before insert

After insert

After update

Before delete

## d. Temporal Triggers

# *List all the temporal triggers and how they will be used to enforce the database constraints*

## e. Procedures

# *List all the procedures and how they will be used to retrieve data required for reports*

# 

## f. Functions

# *List all the functions and how they will be used to retrieve data required for reports*

# 

## g. Views

# *List all the views and how they can be used to provide various displays of the data for different users*

Sales made

Products produced

Raw material levels

# DDL Statements

***Provide the DDL statements that were used to create the database***

CREATE TABLE `employees` ( `date\_of\_change` timestamp(2) NOT NULL DEFAULT CURRENT\_TIMESTAMP(2) COMMENT 'Records the date and time when the data was manipulated. This will help to keep track of the changes made. The assumption is that no 2 users will change the exact same record at the same time (with a precision of a hundredth of a second, e.g. 4.26 seconds).', `employeeNumber` int NOT NULL, `lastName` varchar(50) DEFAULT NULL, `firstName` varchar(50) DEFAULT NULL, `extension` varchar(10) DEFAULT NULL, `email` varchar(100) DEFAULT NULL, `officeCode` varchar(10) DEFAULT NULL, `reportsTo` int DEFAULT NULL, `jobTitle` varchar(50) DEFAULT NULL, `change\_type` varchar(50) NOT NULL COMMENT 'Records the type of data manipulation that was done, for example an insertion, an update, or a deletion.', PRIMARY KEY (`date\_of\_change`), UNIQUE KEY `date\_of\_change\_UNIQUE` (`date\_of\_change`)

CREATE TRIGGER TRG\_BEFORE\_UPDATE\_ON\_employees BEFORE UPDATE ON employees FOR EACH ROW INSERT INTO `classicmodels`.`employees\_undo` SET `date\_of\_change` = CURRENT\_TIMESTAMP(2),

`employeeNumber` = OLD.`employeeNumber` , `lastName` = OLD.`lastName` , `firstName` = OLD.`firstName` , `extension` = OLD.`extension` , `email` = OLD.`email` , `officeCode` = OLD.`officeCode` , `reportsTo` = OLD.`reportsTo` , `jobTitle` = OLD.`jobTitle` , `change\_type` = 'An update DML operation was executed';