

**SOFTWARE REQUIREMENTS  
SPECIFICATIONS DOCUMENT  
FOR EDEN CONFECTIONERIES  
MANAGEMENT SYSTEM.**

## **INTRODUCTION.**

The Purpose of this Section is to describe the System software Requirements Specifications for the Eden Confectioneries Management System which will help to improve on the efficiency and overall effectiveness in management of the organization basing on the Fact finding survey that was carried out by our group.

## **EXTERNAL INTERFACES.**

This addresses how the different types of users Interact with the Eden Confectioneries Management System(ECMS)

### **Types of User Interfaces.**

Customer Interface.

Eden ECMS Login Interface

Management Interface

Custom Order Interface

## **FUNCTIONS.**

These are Categorized as Functional and Non Functional Requirements. Functional Requirements specify the processes and operations that the ECMS should do to solve the problems while Non functional describe the x-tics of the ECMS system.

### **FUNCTIONAL REQUIREMENTS.**

These are categorized into processes, inputs and outputs.

#### **System inputs.**

The system should be able to allow the top managers to post information about readily available bakery products with the various information about them that is price, description, images for customers to make orders.

The system should allow customers to make orders as well as providing entry for information about the order such as date of delivery, location of delivery, contact of customer.

The system should allow customers to make custom orders, in other words, an alternative of making an order by describing it for example providing desired colors, words on the product,flavor,design, shape as the customer may have a preference and the readily available products posted on the site don't match it.

The system should be able to allow the manager to store/enter payment Information about different orders that is the amount paid off or the deposit.

The System should allow customers send social media messages to the organization's social media pages.

#### **System Processes.**

The system should count and keep track of the number of orders per a given time that is weekly, monthly and years.

The system should categorize orders to be delivered in similar or the same geographical location so that they are worked on and delivered at the same time to save transport costs of going to the same place more than once.

The system should be able to evaluate the number of orders that are actually fully paid and cashed, orders that are made through the system but not cashed/picked up at indicated points of delivery.

The system should generate reminders in form of social media notifications to customers who ordered products reminding them of their delivery dates before they actually reach.

### **System Outputs.**

The system should generate a report concerning the company profits according to the initial value invested in buying the bakery raw materials.

The system should generate reminders to managers concerning orders that were paid but whose products were to be picked up or delivered in the near future e.g. a week before the time elapses.

The system should generate a list of frequent customers and their corresponding usual orders so that they can be given occasional discounts to boost market and maintain a healthy customer relationship.

The system should generate warnings when some important information regarding an order is not provided by the customer such as contact, place of delivery and many more.

### **NON FUNCTIONAL REQUIREMENTS.**

These describe how the system behaves to establish constraints of its functionality. They include security, reliability and availability.

#### **Security Requirements.**

The system must ensure that software is protected from unauthorized access to the system and its stored data considering different levels of authorization and authentication across different user roles.

The system must implement a security module to ensure that no user must access restricted system pages through direct URL insertion, that is the system should always redirect them to the home or login page for proper authorized access.

#### **Reliability Requirements.**

The system must have the ability to work without failure for a given period of time. Reliability normally reduces due to bugs in code, hardware problems and other system components.

The system must measure reliability by counting percentage of operations that one completed correctly or track average time system runs before failing.

#### **Operational Requirements.**

The system should run on various kinds of hardware with different views for those different devices that is PC's, tablets, smartphones etc.

The system should always be available at all times and be able to display notifications in case a given component or feature is currently unavailable.

The system should be able to handle concurrent and simultaneous use by multiple users.

### **USABILITY REQUIREMENTS.**

Usability addresses the fact that a system's interfaces and designs are easy to learn how to use as well as remember how to use. It pertains especially to devices that require infrequent use. Users should not be required to consult the system manual each time they need to use the system.

ECMS Interface design should support the following usability requirements.

The system should be efficient in use that is goals should be easy to accomplish and with few or no user errors.

The system should be intuitive that is interfaces should be easy to navigate i.e. contains buttons, headings and help/error messages are simple to understand.

The Interface is characterized by low perceived workload hence it appears easy to use rather than intimidating, demanding and frustrating as well as limited attempts needed by a user to accomplish a particular task.

### **PERFORMANCE REQUIREMENTS.**

Performance describes the quality of responsiveness of the system to various user interactions with it. Poor performance may lead to negative user experience.

The ECMS has a number of performance requirements.

The system must be able to support a minimum of 50 users simultaneously at a given time instance without system failures.

The system must have a minimum response time of not more than 4 seconds.

The system must be operational 24/7 and year after year that is it should be hosted with a reputable hosting company whose servers are strong enough for this operational demand.

### **LOGICAL DATABASE REQUIREMENTS.**

Logical database Design helps define the business data and information requirements into a database system. Logical database designing creates a model which describes each piece of information that the system needs to keep track of.

The ECMS logical database must be able to access and identify all files within the storage system to operate correctly that is manage content on system and descriptions provided by customers for custom orders.

The logical database must categorize information into groups for easier accessibility to save additional time the computer would take searching each file.

The database must stretch over multiple physical hard disks and information files.

The data storage unit must be single for easier information retrieval purposes.

### **DESIGN CONSTRAINTS.**

Design constraints are limiting parameters or boundaries within which the system is to be designed.

The can be described as non functional requirements which the final product should meet.

Design constraints are categorized in terms of language, framework, databases testing and implementing .

#### **Language.**

The system will be written in web languages like HTML, CSS, and java script for front end developing well as the server side language will be entirely PHP.

#### **Framework.**

The system will be built using the modern web framework called **Laravel**. Laravel provides the basic functionalities like authentication and the developers need to add, enhance and manipulate the framework into the desired system without having to develop from scratch.

#### **Database.**

A web based database server side program like PhpMyAdmin which has MySQL embedded in it will be used by the system to manage content, storage and other resources related with the system.

#### **Testing.**

The system will always be tested through out the developing process to see if the implemented features actually function as required by the requirements. This will be done through hosting the prototypes locally using local services like xampp, wampp and mampp.

Hosting locally will enable constant testing until final prototype will be ready enough to be transferred to online hosting.

Trail hosting may also be used.

### **SYSTEM SOFTWARE ATTRIBUTES.**

These describe the system qualities that the final system must meet and have.

They include reliability, security, portability.

#### **Reliability.**

The system shall never crash or hang other than as the result of an operating system error. The system shall provide graceful degradation in the face of network delays and failures.

#### **Security and privacy.**

The system should govern use of manager login account on the organization's equipment.

#### **Maintenance.**

All code shall be fully documented. Each function commented and all program files shall include comments concerning authorship and date of last change.

#### **Portability.**

The system software shall be designed to run on at least one of the following platforms.

Microsoft Windows.

UNIX.

Apple Macintosh OS.

Linux OS.