



Departmental Store Management System

High Level Design & Low Level Design

The purpose of this document is to provide with a template for documenting both HLD & LLD.

Document Control :

Date	Version	Author	Approver Signature
20-10-2022	V-0.1	Group 2	
26-10-2022	V-0.2	Group 2	

Team Members

Employee ID	Employee Name
46274552	Ashutosh Kumar
46275217	Ashutosh Mahapatro
46275252	Azmeera Yamini
46275218	Chintamneedi Aswini
46275309	Ekta Pandey

1. INTRODUCTION	5
1.1. INTENDED AUDIENCE	5
1.2. ACRONYMS/ABBREVIATIONS	5
1.3. PROJECT PURPOSE	5
1.4. KEY PROJECT OBJECTIVES	5
1.5. PROJECT SCOPE AND LIMITATION	6
1.5.1. <i>In Scope</i>	6
1.5.2. <i>Out of scope</i>	6
1.6. FUNCTIONAL OVERVIEW	6
1.7. ASSUMPTIONS, DEPENDENCIES & CONSTRAINTS	6
1.8. RISKS	7
2. DESIGN OVERVIEW	7
2.1. DESIGN OBJECTIVES	7
2.1.1. <i>Recommended Architecture</i>	7
2.2. ARCHITECTURAL STRATEGIES	7
2.2.1. <i>Design Alternative</i>	8
2.2.2. <i>Reuse of Existing Common Services/Utilities</i>	8
2.2.3. <i>Creation of New Common Services/Utilities</i>	8
2.2.4. <i>User Interface Paradigms</i>	8
2.2.5. <i>System Interface Paradigms</i>	8
2.2.6. <i>Error Detection / Exceptional Handling</i>	8
2.2.7. <i>Memory Management</i>	8
2.2.8. <i>Performance</i>	9
2.2.9. <i>Security</i>	9
2.2.10. <i>Concurrency and Synchronization</i>	9
2.2.11. <i>Housekeeping and Maintenance</i>	9
3. SYSTEM ARCHITECTURE	9
3.1. SYSTEM ARCHITECTURE DIAGRAM. (NOT NECESSARY)	10
3.2. SYSTEM USE-CASES	10
3.3. SUBSYSTEM ARCHITECTURE	11
3.4. SYSTEM INTERFACES	11
3.4.1. <i>Internal Interfaces</i>	11
3.4.2. <i>External Interfaces</i>	11
4. DETAILED SYSTEM DESIGN	12
4.1. KEY ENTITIES	13
4.2. DETAILED-LEVEL DATABASE DESIGN	13
4.2.1. <i>Data Mapping Information</i>	16
4.2.2. <i>Data Conversion</i>	16
4.3. ARCHIVAL AND RETENTION REQUIREMENTS	17
4.4. DISASTER AND FAILURE RECOVERY	17
4.5. BUSINESS PROCESS WORKFLOW	17
4.6. BUSINESS PROCESS MODELING AND MANAGEMENT (AS APPLICABLE)	17
4.7. BUSINESS LOGIC	17
4.8. VARIABLES	17
4.9. ACTIVITY / CLASS DIAGRAMS (AS APPLICABLE)	17
4.10. DATA MIGRATION	17

4.10.1. Architectural Representation	17
4.10.2. Architectural Goals and Constraints	17
4.10.3. Logical View	17
4.10.4. Architecturally Significant Design Packages	17
4.10.5. Data model	17
4.10.6. Deployment View	18
5. ENVIRONMENT DESCRIPTION	18
5.1. TIME ZONE SUPPORT	18
5.2. LANGUAGE SUPPORT	18
5.3. USER DESKTOP REQUIREMENTS	18
5.4. SERVER-SIDE REQUIREMENTS	18
5.4.1. Deployment Considerations	18
5.4.2. Application Server Disk Space	18
5.4.3. Database Server Disk Space	18
5.4.4. Integration Requirements	18
5.4.5. Jobs	19
5.4.6. Network	19
5.4.7. Others	19
5.5. CONFIGURATION	19
5.5.1. Operating System	19
5.5.2. Database	19
5.5.3. Network	19
5.5.4. Desktop	19
6. REFERENCES	19
7. APPENDIX	19

1. Introduction

DSMS stands for **D**epartmental **S**ore **M**anagement **S**ystem. The term is applicable to a retail shop or store, the checkout/cashier counter in the store, or a location where such transactions can occur in this type of environment. Departmental store management system is utilized in many different industries, ranging from restaurants, hotels and hospitality businesses, nail/beauty salons, casinos, stadiums etc. In the most basic sense, if something can be exchanged for monetary value - a Departmental Store Management System can be used.

1.1. Intended Audience

This document is intended to read by developers, testers, project managers and customers. This is a technical document, and the terms should be understood by all of them. It is a platform to organize their products and Customers in an orderly manner and dynamically make changes to the Departmental Store as per the user's criteria and requirements.

1.2. Acronyms/Abbreviations

DSMS	Departmental Store Management System
------	--------------------------------------

1.3. Project Purpose

The Purpose of this project and documentation is to build a Departmental Store Management System using C Programming and its various supporting tools to add, edit, delete and view product and customer records. It allows you to reduce pricing errors and speed up checkouts. This also enables cashiers to process transactions and serve customers efficiently and allows managers to maintain tight control.

1.4. Key Project Objectives

- ❖ Main objective is to store and manage the details of Products customer's.
- ❖ Users can add customer and product records.
- ❖ We can edit and delete the records.
- ❖ We can display all the list of products stored.
- ❖ To reduce the manual work for managing the stocks, products and sales transactions etc.
- ❖ It tracks all the details about payments, sales, and products.

1.5. Project Scope and Limitation

The scope of this project is to create the departmental store system to assist any organization in storing and retrieving all the information about a preexisting customer in a way more robust and efficient manner. All the information about a particular customer is stored in a retrievable manner.

1.5.1. In Scope

The departmental store provides all facilities under one roof to enable the customer to by all his requirements. Large variety of products are stored. This has a special appeal to the customers. The end goal is to create an application that is easy to use, understand and respond to user queries in a fluid manner

1.5.2. Out of scope

No other person like customers or third-party persons cannot access this application. Only admin or departmental employee can access it for storing, editing, or deleting product records.

1.6. Functional Overview

The Departmental Store management system is developed by using C Programming language. This application allows the user to perform operations like storing product records, editing the records, deleting records etc. This application is a console application without graphics. File handling is used in while developing this application. This application is solved using several methods, like one can solve this program using user defined function concept, loop condition and conditional statements.

The Departmental store management application tracks your sales and product information. It is a file for the communications you make daily including product name, product code, product quantity, billing counter and customer details. You can use customer details and product details to easily organize and find all the information linked to departmental store. The application integrates al your product information for different orders. The main purpose of departmental store management is to record and organize your product and customer details.

1.7. Assumptions, Dependencies & Constraints

The following assumptions have been made in regards to the development of the Departmental Store Management System.

- The user or client organization has machines capable of running a Linux based operating system.

- User has the latest version of Ubuntu Linux.
- Client has either an 4GB or more RAM.
- The service is used preferably on a desktop or laptop.
- C source code can be compiled on the machines.
- The users have some storage space to store the contact details

1.8. Risks

- Irrelevant contacts should not be displayed during searches to protect privacy.
- The implementation should not be susceptible to easy modification
- The source code needs to be implemented in such a way that it is portable to any machine that can compile and run C programs.

2. Design Overview

Operating Environment for the Departmental Store Management System is as follows:

- Operating System: Any Linux based operating system.
- Compiler: GCC or Similar source code written in C programming language.

2.1. Design Objectives

- Users can add product and customer records and can search for the product.
- All records can be displayed in an organized manner.
- There are functions to add, edit, delete and to display the records based on user input.
- We can show sales report also.

2.1.1. Recommended Architecture

The recommended architecture is as follows

- 4GB RAM
- Internet connection
- Desktop or Linux machine
- Internet connection
- Terminal

2.2. Architectural Strategies

- Add product record: This allows the user to add a product record to the database.
- Edit product record: This will allow us to edit the product name and quantity based on product code.
- View product record: This will allow us to see all the product records we have stored.

- Delete product record: This will allow us to delete the record based on product code.
- Add customer record: This will allow us to insert customer name, age, product code, product name etc.
- Sales report: This will display the daily sales report.
- Hotcake report: This will display the product that is maximum sold in a day.

2.2.1. Design Alternative

Not Applicable

2.2.2. Reuse of Existing Common Services/Utilities

Not Applicable

2.2.3. Creation of New Common Services/Utilities

The project does not create or use any new common services or utilities.

2.2.4. User Interface Paradigms

- Desktop or a Linux machine with internet connection.
- Command Line Interface (CLI).

2.2.5. System Interface Paradigms

- Operating system – Linux.
- Linux Kernel version – 6.0.3
- Bash shell: x86_64 GNU/Linux

2.2.6. Error Detection / Exceptional Handling

- This software is completely bug free and error free. The exceptions are handled during making of this document.

2.2.7. Memory Management

We are storing the data in a text file.

2.2.8. Performance

Taking the essential speediness into consideration we will be using this departmental Store management System for quick storing and retrieving. System design will fulfill performance requirements.

2.2.9. Security

The software system needs a robust security mechanism in place so that unauthorized users are not allowed access to Departmental store management system. All users of the system must be uniquely identified. This could be done by using a user name and associated password scheme that would authenticate and authorize the user access to the system and, if applicable, grant the user access to restricted or controlled Departmental store management system. If a user cannot be identified, he/she will be given “anonymous” access with read-only capabilities. In order to monitor all past access to the system, all attempts to access the system must be logged.



All users of the system shall login using some form of unique identification (e.g., username and password). All login attempts shall be done so in a secure manner (e.g., encrypted passwords). Each user shall either be trusted or not trusted.

2.2.10. Concurrency and Synchronization

Not Applicable

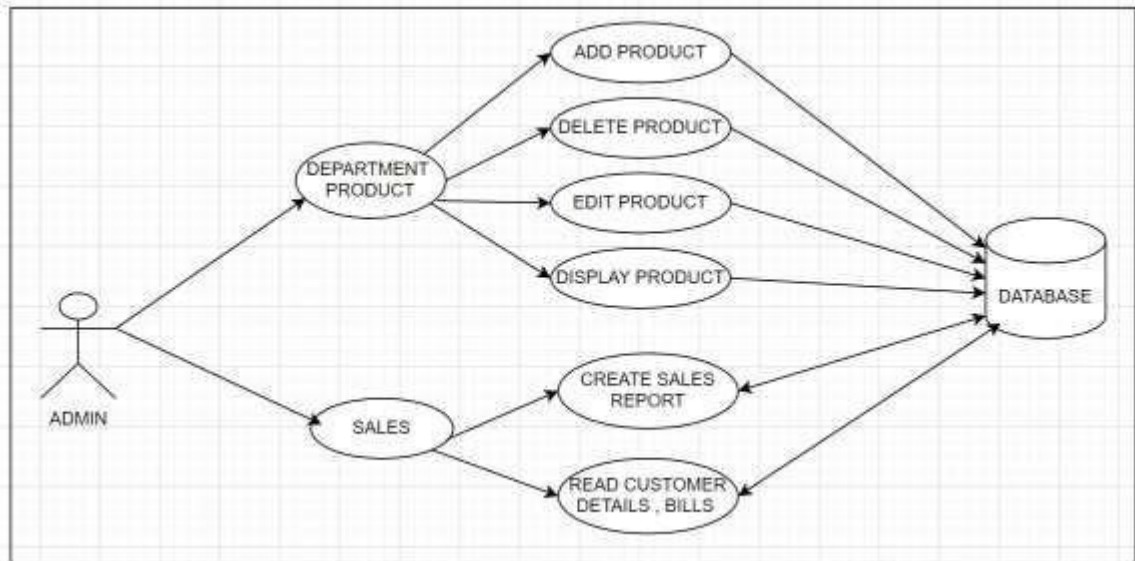
2.2.11. Housekeeping and Maintenance

Not Applicable

3. System Architecture

The below is the system architecture diagram for our project Departmental Store Management System.

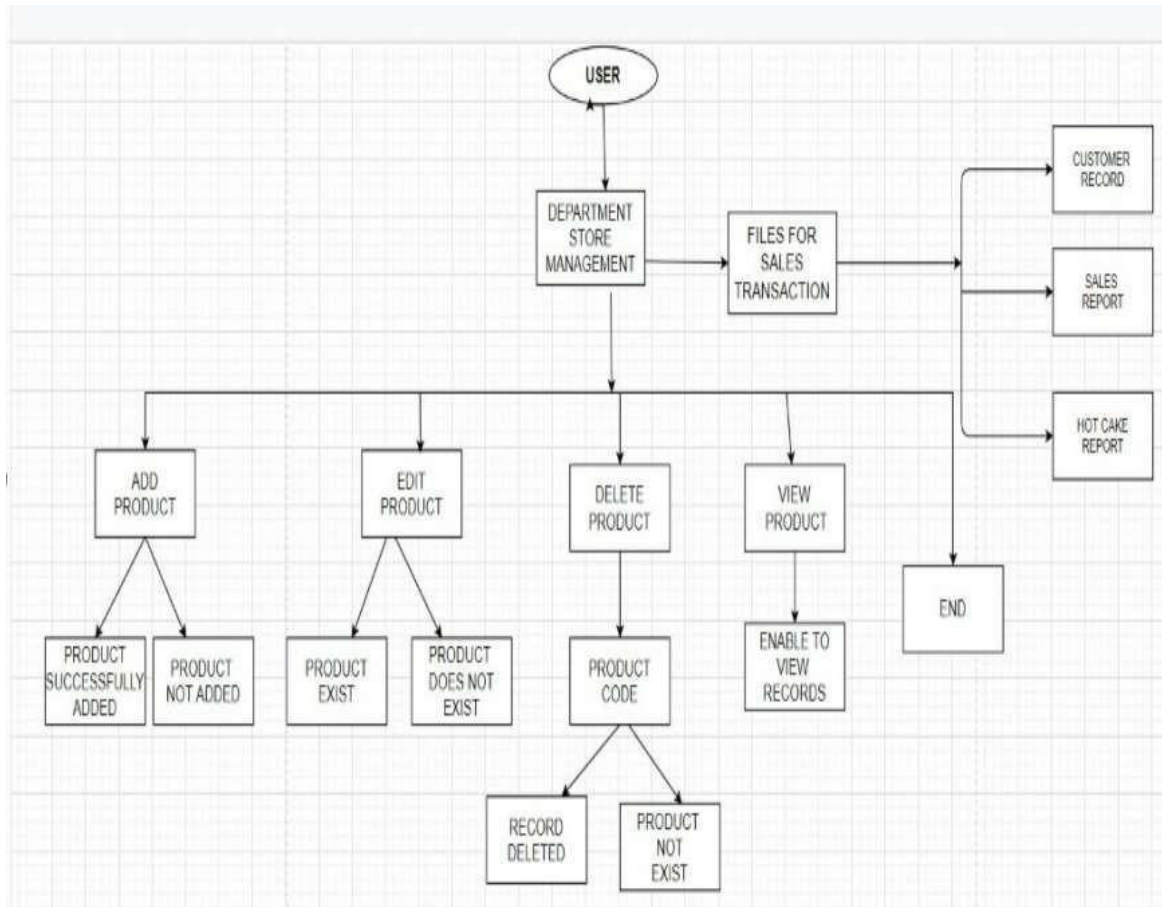
3.1. System Architecture Diagram.



3.2. System Use-Cases

Not Applicable

3.3. Subsystem Architecture



3.4. System Interfaces

Different interfaces are given below

3.4.1. Internal Interfaces

1. Language - C Programming

3.4.2. External Interfaces

1. Server Configuration: Minimum 2GB Hard Disk
2. P-III processor or equivalent
3. Ram 512 MB
4. Linux
5. Client Configuration

3.5 Functions Used

- 3.5.1. isCodeAvailable() :- This function will check the product exist or not.
- 3.5.2. isProductAvailable() :- This function will check that the product is available in stock or not.
- 3.5.3. getChoice() :- This function will give choice to choose.
- 3.5.4. checkRate() :- This function helps to check the rate of the product.
- 3.5.5. addProduct() :- This function will help to add product in the stock.
- 3.5.6. displayProduct() :- This function will help to display the product.
- 3.5.7. closeApp() :- This function will close the app.
- 3.5.8. searchProduct() :- This function will help to search the product from the stock.
- 3.5.9. deleteProduct() :- This function will help to delete product from the stock.
- 3.5.10. updateProduct() :- This function will update the details of the products.
- 3.5.11. login() :- This function is used when user want to login.
- 3.5.12. addCustomer() :- This function is used when user want to add customer.
- 3.5.13. viewCustomer() :- This function is used when user want to view the customers.
- 3.5.14. mainMenu() :- This function is at where we have login function.
- 3.5.15. saleProduct() :- This function is used when user sells the product.
- 3.5.16. options() :- This function is help us to choose to what different function of products like add product, delete product etc.
- 3.5.17. customerSalesReport() :- This function take details from user and reflects the report.

4. Detailed System Design

The Departmental Store Management System (DSMS) can be handled by admin or departmental store employee.

Admin can add product record to the file. He asks for the product code, product name, units of product available and zone code of whose record to be added. Upon successful addition of new product record to the file, it will show the message of "Product Successfully added". If the department employee will give blank product name, then it will show the message of "Product name should not be blank".

Admin can also edit the added product record like product name, product quantity based on product code. The product code which is entered by employee will not found in record file, then it will show the message of "Product does not exist".

Admin can also delete the product record from the file. For this employee has to enter the product code to delete the record. Upon successful deletion of product, it will show the message of "Record deleted Successfully". And also, he can display all records in the file at once.

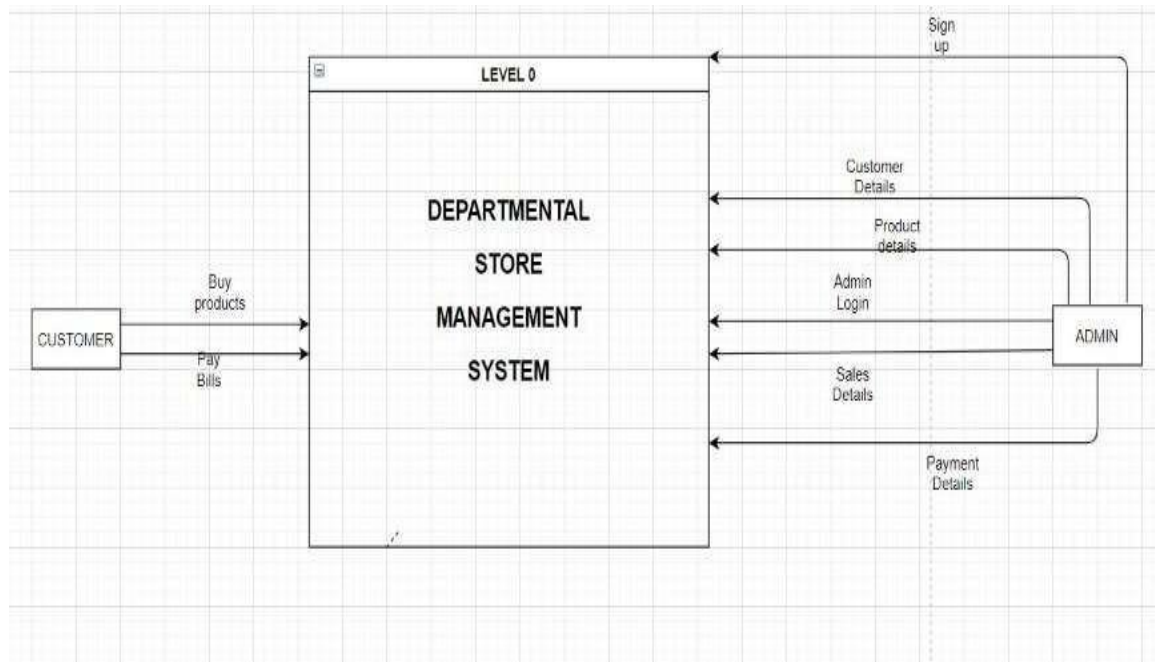
Departmental employee can add new customer record to the file. It asks for customer name, age, product code, quantity of product. He can also show sales report and the maximum product in day.

4.1. Key Entities

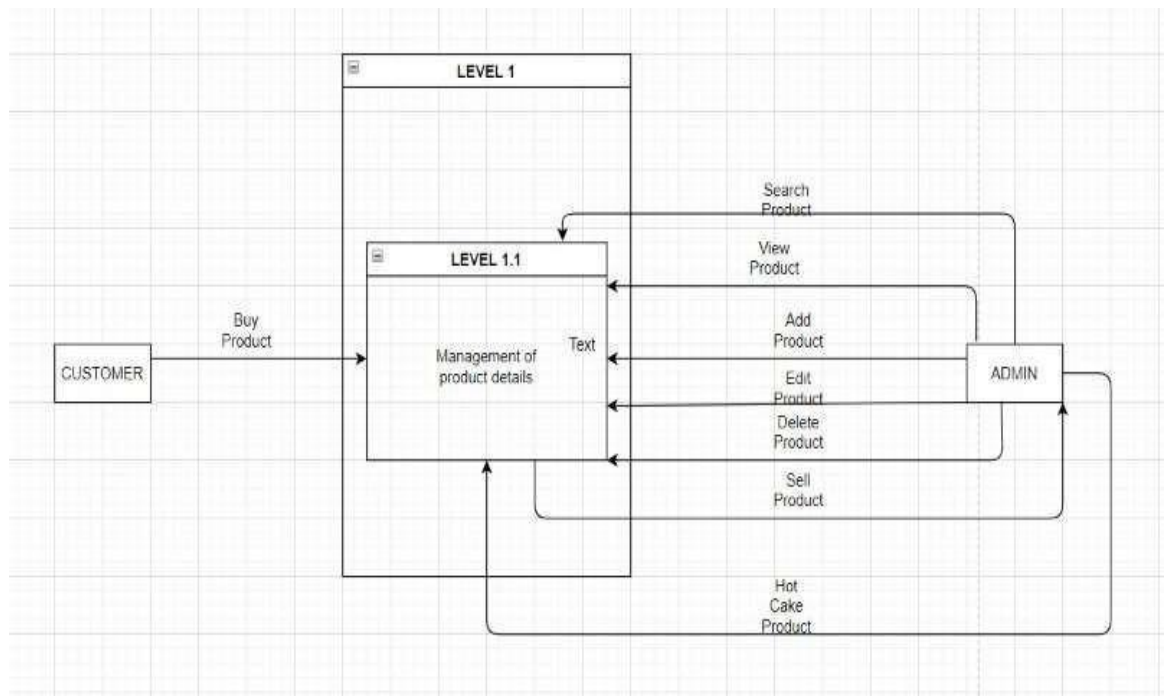
Not Applicable

4.2. Detailed-Level Database Design

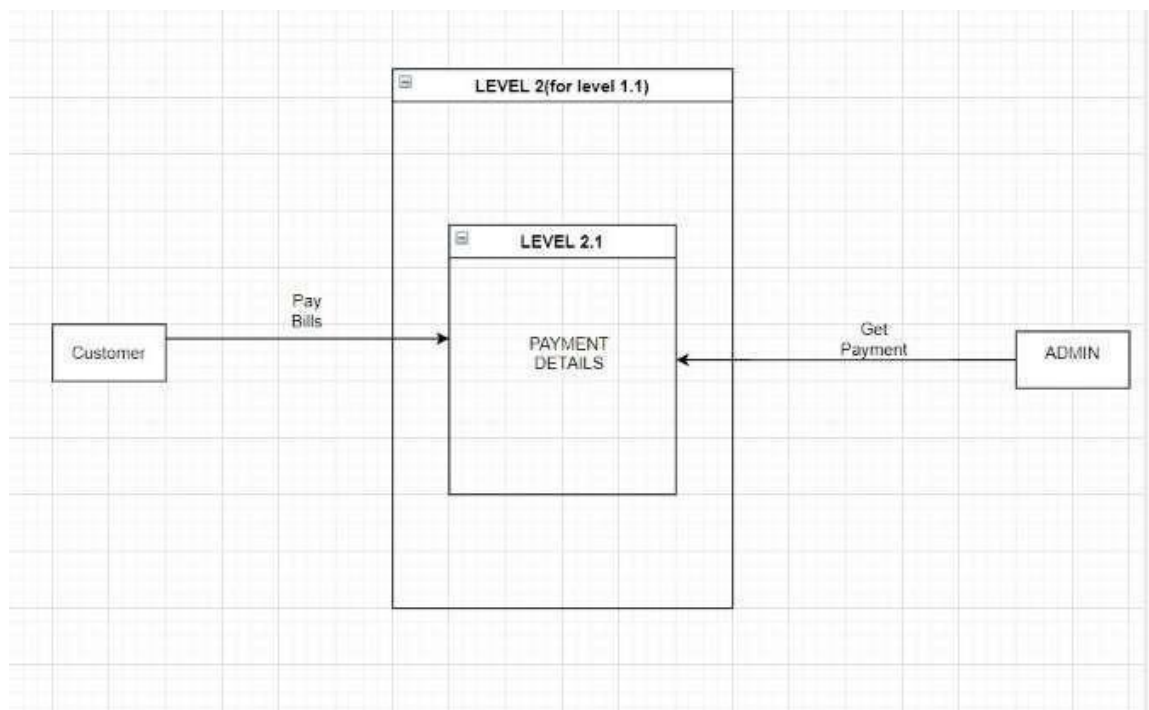
Level 0 Diagram



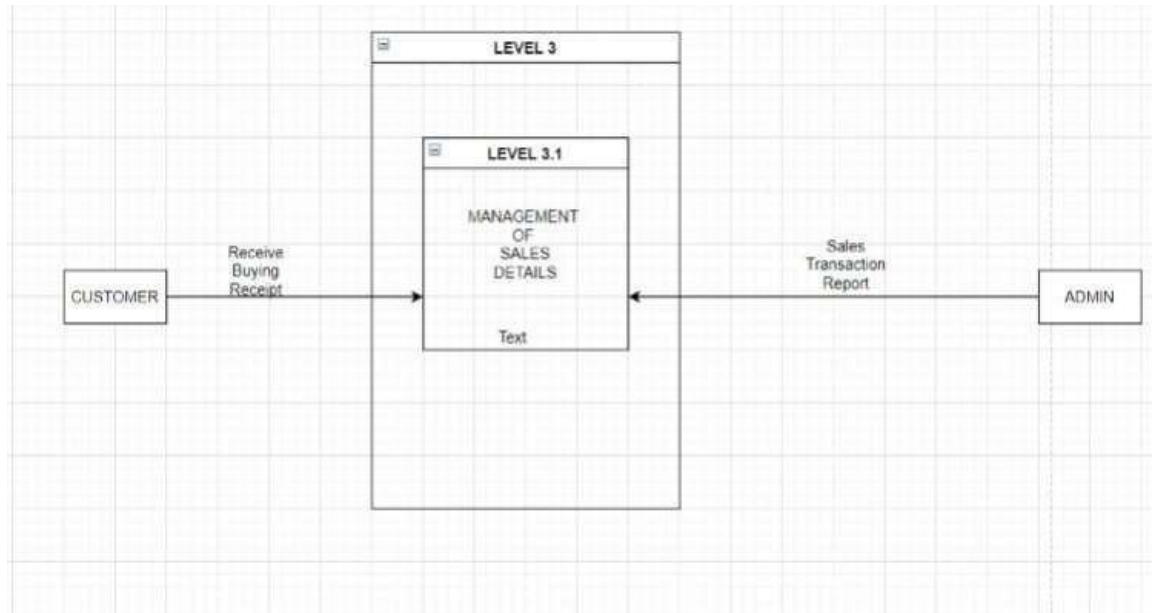
Level 1 Diagram



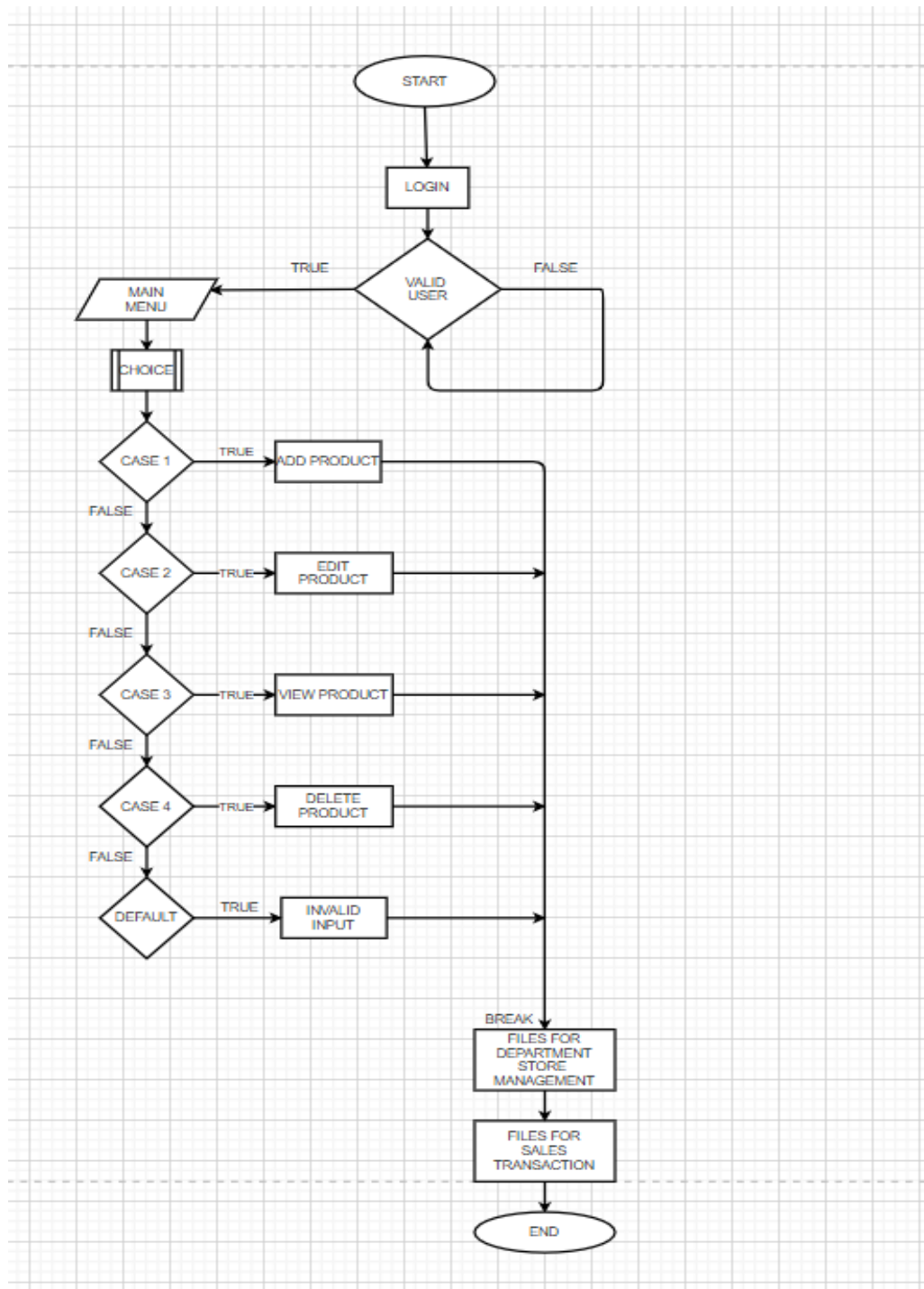
Level 2 Diagram



Level 3 Diagram



4.2.1. Data Mapping Information



4.2.2. Data Conversion

Not Applicable

4.3. Archival and retention requirements

Not Applicable

4.4. Disaster and Failure Recovery

Not Applicable

4.5. Business Process workflow

Not Applicable

4.6. Business Process Modeling and Management (as applicable)

Not Applicable

4.7. Business Logic

Not Applicable

4.8. Variables

Not Applicable

4.9. Activity / Class Diagrams (as applicable)

Not Applicable

4.10. Data Migration

Not Applicable

4.10.1. Architectural Representation

Not Applicable

4.10.2. Architectural Goals and Constraints

- Can keep all your customer information in one place.
- Can keep all product details in one place.
- It keeps customer purchase history.
- Build your own reports and quick see the store.

4.10.3. Logical View

Not Applicable

4.10.4. Architecturally Significant Design Packages

Not Applicable

4.10.5. Data model

Not Applicable

Legacy system data model

Proposed system data model
Interface data model

4.10.6. Deployment View

Not Applicable

5. Environment Description

Environment description for the system is given below.

5.1. Time Zone Support

It will support time zone as per Indian standard time (IST) in (GMT +5:30) and UST standard.

5.2. Language Support

C language and compilation using Gcc. The Linux commands to do that task we can specify the commands.

5.3. User Desktop Requirements

User desktop requires a Linux environment, Operating system of Linux Debian or Terminal x86_64 GNU/Linux and kernel version 4.4.0-19041-Microsoft #1237Microsoft and reliable internet connectivity.

5.4. Server-Side Requirements

Not Applicable

Deployment Considerations

Deployment considerations are,

- 500Mhz Processor
- 120GB HDD CPU
- 4GB RAM
- Network connectivity

5.4.1. Application Server Disk Space

Disk space – Minimum 150GB.

5.4.2. Database Server Disk Space

Not Applicable

5.4.3. Integration Requirements

The PWD Command Displays the current working directory on the server for the logged in user.

5.4.4. Jobs

Not Applicable

5.4.5. Network

The network connects the system for the purpose of file searching therefore stable Internet connectivity is required.

5.4.6. Others

Not Applicable

5.5. Configuration

5.5.1. Operating System

- Operating system – Linux.
- RAM - 4GB.
- Processor - i3/i5/i7.

5.5.2. Database

In our project we are using file handling in C to store details.

5.5.3. Network

Must have good internet connectivity.

5.5.4. Desktop

Linux like environment is required.

6. References

- http://www.academia.edu/6967044/Software_Requirements_Specification_for_Point_Of_Communication_Sale_System (Last Access : 22/02/2017)
- System Requirements Specification Document
- <https://www.geeksforgeeks.org/basics-file-handling-c/>
- <https://www.geeksforgeeks.org/how-to-use-make-utility-to-build-c-projects/#:~:text=Makefile%20is%20a%20set%20of,of%20times%20by%20using%20Makefile.>

7. Appendix

Not Applicable

Change Log

OMS Temnlate Version Control

Date	Version	Author	Description
28-May-2015	1.0	QA Team	Initial Version