Capgemini

High Level Design & Low Level Design

Document Control :

Project Revision History

Date	Version	Author	Brief Description of Changes	Approver Signature
29.08.2022	1.0	Group 6		

Index

Introduction: -

1.1 Intended Audience: -

There is no particular audience set for this project as anyone who wants to buy a product from departmental store.

1.2 Project Purpose: -

The introduction of the software requirement specification provides an overview of the entire software. This is an overview description purpose, scope, tools used and basic description. The aim of this document is to gather, analyze and give an in-depth insight into the complete Departmental Store Management System by defining the problem statement in detail.

The detailed requirements of the Departmental Store System are provided in this document.

1.3 Key Project Objectives: -

a Allow departmental employee to add record of product to file.
b Modify/Update the added records.
c Delete record from file
d Allow user to buy product
e Generate report of the daily sales
f Generate report of product details

1.4 Project scope and limitation: -

Instead of writing the all the details of the departmental store at the piece of paper or some other place, we can save all of it at a single place. Also, all sort of corrections can made in a proper and neat manner.
As far as limitation is concerned, we can store any other information only at the given in this project.

1.5 Functional Overview: -

1.5.1 Following header files are included in the program:

a #include <stdio.h>
b #include <ctype.h>

```
c   #include <cunit.h>
d   #include <string.h>
e   #include <pthread.h>
f   #include <stdlib.h>
```

1.5.2 Following functions are included in the program:

1.Add_Product() :- Add the new product in the Department Store.

This is the function that is built in the departmental store system to add the new product.It is held responsible for adding the product along with the product name product code and the quantity required for the same

2.Edit_Product() :- Edit all the product in the product in the Department Store.

This is the function which is use to edit the pre existing the records you can edit the product name quantity of the existing record it also asks for the product code initially that is needed to be added and if the product is pre existing it will ask for the updated data for the same product.

3.Delete_Product() :-Delete the product in the Departmental Store.

This is the function that is asked for the product code that is need to be deleted if the product code entered matches with the code of the pre existing product record it ultimately that particular record from the Store.

4.View Product(0:- It views all the details of the files.

5. Create_Queue ():- Stored all the information in a form of queue of the departmental store.

Create queue is held responsible for storing all the details of the customer and the product along with the product code product name unit zone code customer name , age along with certain condition that are given as follows.

- If the age entered by the employee of the customer is between the age group of 15 to 35 it store it in queue 1 and if the age entered of the customer is above 35 it is stored in queue 2.

- Product code should be 3 digit only.
- Customer name and Product name should only contain the alphabets.
- Zone code should be only X,Y,Z.

all other values entered will stop the process at that specific interval of time and declare it as invalid.

6. Dequeue():-

It dequeue the product from the product file and put it in sales transaction files for further the sales.

7. Get_Password():-

It asks for the password before accessing any of the details of the product and the customer for a password that is saved and parellel checks that the password is valid or not, if the password entered is valid then only you can perform all the function in the folder.

8.Start Sale();

It starts the sales of the product after dequeue.

9.Sales_Report()

It display the report of the unit solds of the particular product along with the total sum of the prices of the records.

10.hotcake();

It display the maximum numbers of units that are being sold.

11..Display():-

It display all the data that is stored in the file accordingly.

Department Store comprises of the following modules:

Name of the Module	Get Password
Handled by	Muskan Yadav
Description	It checks that password entered is valid or not for accessing the contents of the department store.

Name of the Module	Add Product
Handled by	Dolly Saluja
Description	The user adds the record in the file

Name of the Module	Edit Product
Handled by	Snehal Girish Bagul
Description	The user edit a record from file

Name of the Module	Delete Product
Handled by	B.Poojitha
Description	The user deletes the record from File

Name of the Module	Dequeue
Handled by	Muskan Yadav
Description	It Dequeue all the relevs

Name of the Module	Create Queue
Handled by	Nidhi Dubey
Description	The user stores the record in the File

Name of the Module	Start Sale
Handled by	Muskan Yadav
Description	It starts the sale.

Name of the Module	View Product
Handled by	Nidhi Dubey
Description	It views all the products.

Name of the Module	Sales Report
Handled by	Dolly Saluja
Description	It display the total unit sold and the product name along with all the relevant details.

Name of the Module	Hotcake Report
Handled by	Snehal Girish Bagul
Description	It display the details of the particular product whose maximum number of unit get solds.

Name of the Module	Display Module
Handled by	B.Poojitha
Description	It will display the data from file.

2.1 Design Objectives: -

- Add product
- Edit product
- Delete product
- View Product
- Sales report
- Hotcake report

2.2 Design Alternative: -

we have used linked list to perform all the relevant operations in the particular file .

2.3 User Interface Paradigms: -

The Departmental Store progives a user an option to have its personal contact diary stored system file. A system always works faster than a person can. User is given an interface to add a new record in case he wants to add a record, an option to delete a record, search a record with various options, update a record, view the contacts.

2.4 Error Detection / Exceptional Handling: -

- User should first enter the details according the condition and if the entered detail is not according the condition specified sometimes it is displays the message that is entered and sometimes it returns with an error.

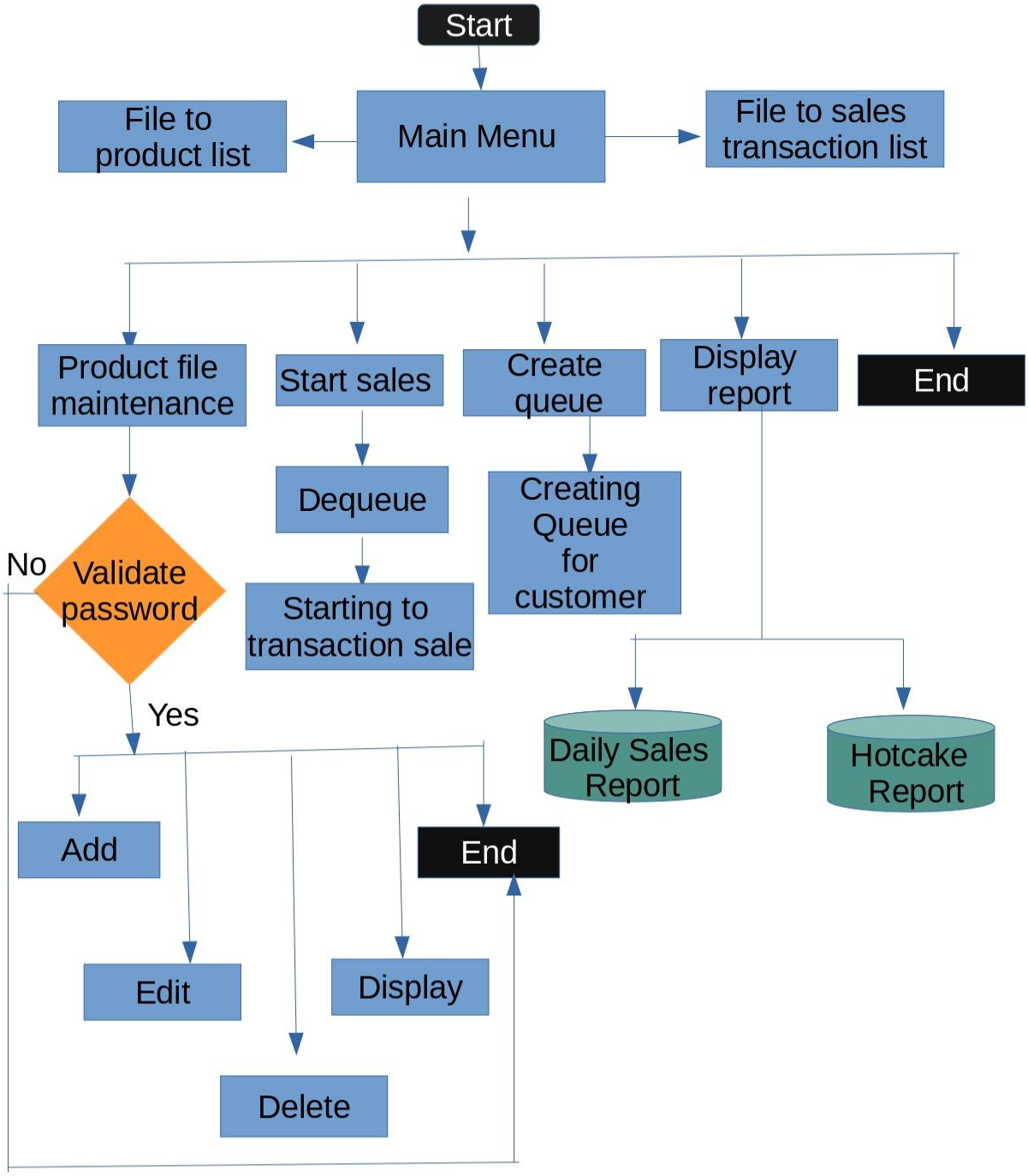
2.5 Performance: -

The performance depends up on the hardware component and cloud working of the user's system.

2.6 Maintenance: -

Very little maintenance should be required for this setup. An initial configuration will be the only system required interaction after system is put together. The only other user maintenance would be any changes to settings after setup, and any specified special cases where user settings or history need to be changed. Physical maintenance on the system's parts may be required, and would result in temporary loss of data or Internet. Upgrades of hardware and software should have little effect on this project but may result in downtime.

DETAILED SYSTEM DESIGN



. Environment Description: -

5.1 Time Zone Support: - IST

5.2 Language Support: - English

5.3 User Desktop Requirements: -

a. 64-bit processor, 1 GHz or faster
b. At least 2 GB free hard drive space
c. At least 1 GB RAM

5.4 Server-Side Requirements: -

a 32 64-bit processor, 1 GHz or faster
b At least 1 GB free hard drive space
c At least 1GB RAM

5.4.1. Deployment Considerations: -

.a Easy setup: no session storage daemon, use tmpfs and memory caching to enhance performance.
.b Local storage is used
.c No network latency to consider
.d To scale buy a bigger CPU, more memory, larger hard drive, or additional hardware

5.4.2. Application Server Disk Space: -

No such disk space is required as the program is fully functional on online IDE(s) as well. Local Operating System is required and one txt file to store the records of processes.

5.4.3. Database Server Disk Space: -

No such disk space is required as the program is fully functional on online IDE(s) as well.

5.4.4. Integration Requirements: -

a. Language: - C
b. Tools: - Gcov, Valgrind, Makefile ,Cunit
c. Complier: - g++
d. Linux Environment

5.4.5. Network: - End to End

5.5 Configuration: -

5.5.1. Operating System: - Linux environment

6. Reference: -

https://www.geeksforgeeks.org/data-structures/linked-list/

https://www.digitalocean.com/community/tutorials/queue-in-c

https://www.geeksforgeeks.org/thread-functions-in-c-c/

https://www.programiz.com/dsa/deque#:~:text=Check%20if%20deque%20is%20empty,to%20the%20front%20front%20%3D%200%20.

http://cunit.sourceforge.net/

https://www.javatpoint.com/c-string-functions

https://gcc.gnu.org/onlinedocs/gcc/Gcov.html

https://www.tutorialspoint.com/unix_commands/gprof.htm#:~:text=Gprof%20reads%20the%20given%20object,in%20the%20given%20profile%20files.

https://www.tutorialspoint.com/makefile/index.htm#:~:text=Makefile%20is%20a%20program%20building,help%20of%20user%2Ddefined%20makefiles.