



High Level Design & Low Level Design

Document Control:

Project Revision History

Date	Version	Author	Brief Description of Changes	Approver Signature
27.09.2022	1.0	Group 4		

Index

1. Introduction	5
1.1 Intended audience	5
1.2 Project purpose	5
1.3 Key project objective	5
1.4 Project scope	5
1.5 Functional overview	6
1.5.1 Header files	6
1.5.2 Functions	6
2. Design Overview	13
2.1 Design objective	13
2.2 Design alternative	13
2.3 User interface Paradigms	13
2.5 Validations	13
3. Detailed System Design	14
3.1 Flow Chart	14
3.2 DFD of Level 0	15
3.3 DFD of Level 1	15
4. UML Diagram	16
4.1 Use Case Diagram	16
4.2 Sequence Diagram	17
4.3 Class Diagram	18
4.4 ER Diagram	19
5. Environment description	19
5.1 Time zone support	19
5.2 Language support	19
5.3 User desktop requirement	19
5.4 Server-side requirements	19
5.4.1 Deployment consideration	19
5.4.2 Database server disk space	20

5.4.3 Integration requirements	----- 20
5.4 Configuration	----- 20
5.4.1 Operating System	----- 20
6. Experimental Results	-----21
6.1 Unit testing	-----21
6.2 Integration testing	-----23
7. Tools Report	-----26
7.1 Gprof	-----26
7.2 Gcov	-----27
7.3 CppCheck	-----28
7.4 Valgrind	-----28
8. Reference	----- 28

1. Introduction: -

1.1 Intended Audience: -

The audience set for this project would include admin who handles the system, suppliers are the vendors that provide products and customer is the one who will purchase IT-Hardware items.

1.2 Project Purpose: -

The online IT-Hardware ordering system is specially designed for online ordering applications with facilities of order placing, creating account, bill generation etc. Secondly, the application also allows only the authorised employee to maintain the supplier, maintain customer and maintain the product. The system also allows employees to view top 3 sold products, transactions done by the customer, and order a stock if stock is empty. Whereas the customer and supplier use this system to create an account, edit and view the details.

1.3 Key Project Objectives: -

- a. Allow the customer and supplier to login into the portal.
- b. The application allows the customer to place the order.
- c. The application allows the customer to order a maximum of 3 products.
- d. The system will display the bill to the customer.
- e. Allows the supplier to add and remove the product.
- f. A stock database is maintained for all the products.
- g. Admin can access both customer and supplier data.

1.4 Project scope: -

This application can be implemented in any IT-Hardware ordering store in the locality which can accept orders 24/7. In the future, we can add a home delivery feature which will make our project more flexible. With the help of our system, the customer can easily purchase the items without any delay.

1.5 Functional Overview: -

1.5.1 Following header files are included in the program:

- a. `#include <iostream>`
- b. `#include <iomanip>`
- c. `#include <vector>`
- d. `#include <string>`
- e. `#include <list>`
- f. `#include <limits>`
- g. `#include "header.hpp"`
- h. `#include <thread>`
- i. `#include <mutex>`

1.5.2 Following functions are included in the program:

1. Admin Module: The admin module is one of the main components of the Zegat Online IT-Hardware Ordering system. Admin will have login details and after entering the login details he/she can access the features. Admin module provides the following features:

- **edit_supplier():** Admin can edit the details of any supplier like their Name, mobile number, and address.
- **remove_supplier():** Admin can remove any supplier from the list.
- **view_supplier_list():** Admin can view the list of all the suppliers.
- **edit_customer():** Admin can edit the details of any customer like their Name, mobile number, and address.
- **remove_customer():** Admin can remove any customer from the list.
- **view_customer_list():** Admin can view the list of all the customers.
- **add_product():** Admin can add the product to any supplier by entering the supplier ID.
- **edit_product():** Admin can edit the details of any product by entering the product ID.
- **remove_product():** Admin can remove any product by entering the product ID.
- **view_product_list():** Admin can view the list of all the products.
- **top_3_product():** Admin can view the top 3 sold products.
- **order_stock():** If the stock is out of range, then the admin can order the stock.
- **stock_report():** In this function, it will display the total stock.

- **view_transactions():** Admin can view transaction reports on a daily basis.

2. Supplier Module: A supplier module is one of the main modules of the Zegat Online IT-Hardware Ordering system which provides the following features:

- **register_supplier():** The supplier can register by providing his/her basic info like name, address, or Mobile no. etc on our Zegat Online IT-Hardware Ordering System.
- **login_supplier():** The supplier can log in using the supplier Id and can access the supplier features.
- **edit_supplier_details():** The supplier can edit his basic info like his name, address, or mobile no. etc and it will get updated into the supplier database.
- **view_supplier():** The supplier can view the details entered by him.
- **products_supplied():** This function will give the product which is supplied to the customer.
- **add_product():** Supplier can add the product by giving the information like product name, units, cost of each item.
- **edit_product():** The supplier can edit the product after the product is supplied.

3. Customer Module: The customer module provides the following features.

- **register_customer():** New customers can register by providing details like Name, address, and mobile number.
- **login_customer():** Customers can log in by entering their customer ID to continue shopping.
- **view_transaction_history():** Customers can view all their previous transactions by this function.
- **edit_customer_details():** Customers can edit their details like Name, mobile number, and details.
- **view_items():** The customer can view all the items which are supplied by the supplier and can order from them.

- **place_order():** The customer can place an order by entering the product ID and the units he wants to purchase. After placing the order, a bill will be generated.
- **view_customer():** The customer can view the details of himself.

Zegat Online IT-Hardware Ordering System comprises three modules and the following member functions:

Name of the Module	Main menu
Handled by	Twinkle Jain
Description	It consists of the main menu and sub-menus which will provide options to the user.

Name of the Module	register_customer()
Handled by	Vaidehee Dalvi
Description	New customers can register by providing details like Name, address, and mobile number.

Name of the Module	login_customer()
Handled by	Vaidehee Dalvi
Description	Customers can login by entering their customer ID to continue shopping.

Name of the Module	view_customer()
Handled by	Vaidehee Dalvi
Description	The customer can view the details of himself.

Name of the Module	edit_customer_details()
Handled by	Vaidehee Dalvi
Description	Customers can edit their details like Name, mobile number, and details.

Name of the Module	view_items()
Handled by	Vaidehee Dalvi
Description	Customers can view all the items which are supplied by the supplier and can order from them.

Name of the Module	place_order()
Handled by	Vaidehee Dalvi
Description	The customer can place an order by entering the product ID and no of units he wants to purchase. After placing the order, a bill will be generated.

Name of the Module	view_transaction_history()
Handled by	Vaidehee Dalvi
Description	Customers can view all their previous transactions by this function.

Name of the Module	register_supplier()
Handled by	Mohd Uzair Khan
Description	The supplier can register by providing his/her basic info like name, address, Mobile no. etc on our Zegat Online IT-Hardware Ordering System.

Name of the Module	login_supplier()
Handled by	Mohd Uzair Khan
Description	The supplier can login using the supplier Id and can access the supplier features.

Name of the Module	edit_supplier_details()
Handled by	Mohd Uzair Khan

Description	The supplier can edit his basic info like his name, address, or mobile no. etc and it will get updated into the supplier database.
-------------	--

Name of the Module	view_supplier()
Handled by	Mohd Uzair Khan
Description	The supplier can view the details entered by him.

Name of the Module	products_supplied()
Handled by	Mohd Uzair Khan
Description	This function will give the product which is supplied to the customer.

Name of the Module	add_product()
Handled by	Mohd Uzair Khan
Description	Suppliers can add the product by giving information like product name, units, and cost of each item.

Name of the Module	edit_product()
Handled by	Mohd Uzair Khan
Description	Suppliers can edit the product after the product is supplied.

Name of the Module	remove_supplier()
Handled by	Twinkle Jain
Description	Admin can remove any supplier from the list.

Name of the Module	remove_customer()
Handled by	Twinkle Jain
Description	Admin can remove any customer from the list.

Name of the Module	remove_product()
Handled by	Girisha M
Description	Admin can remove any product by entering the product ID.

Name of the Module	view_supplier_list()
Handled by	Twinkle Jain
Description	Admin can view the list of all the suppliers.

Name of the Module	view_customer_list()
Handled by	Twinkle Jain
Description	Admin can view the list of all the customers.

Name of the Module	view_product_list()
Handled by	Girisha M
Description	Admin can view the list of all the products.

Name of the Module	view_transactions()
Handled by	Vaishali Jain
Description	Admin can view transaction reports on a daily basis.

Name of the Module	add_product()
Handled by	Girisha M
Description	Admin can add the product to any supplier by entering the supplier ID.

Name of the Module	edit_supplier()
Handled by	Twinkle Jain

Description	Admin can edit the details of any supplier like their Name, mobile number, and address.
-------------	---

Name of the Module	edit_customer()
Handled by	Twinkle Jain
Description	Admin can edit the details of any customer like their Name, mobile number, and address.

Name of the Module	edit_product()
Handled by	Girisha M
Description	Admin can edit the details of any product by entering the product ID.

Name of the Module	order_stock()
Handled by	Vaishali Jain
Description	If the stock is out of range, then the admin can order the stock.

Name of the Module	top_three_product()
Handled by	Girisha M
Description	Admin can view the top 3 sold products.

Name of the Module	stock_report()
Handled by	Vaishali Jain
Description	In this function, it will display the total stock.

2. Design Overview

2.1 Design Objectives: -

- Creating a user-friendly Zegat Online IT-Hardware Ordering Application.
- Customers will login into the application and view all the items.
- Customers can place an order and view the transaction.
- Products will be supplied by the supplier and if needed, the supplier can add, edit, and view the products.
- Admin maintains both the customer and supplier like remove, view, add or edit.
- The top 3 most sold products can be displayed by the admin.
- The stock report is also generated to view the stock.

2.2 Design Alternative: -

We have used Standard Template Library (STL) to perform all the relevant operations in the particular file.

2.3 User Interface Paradigms: -

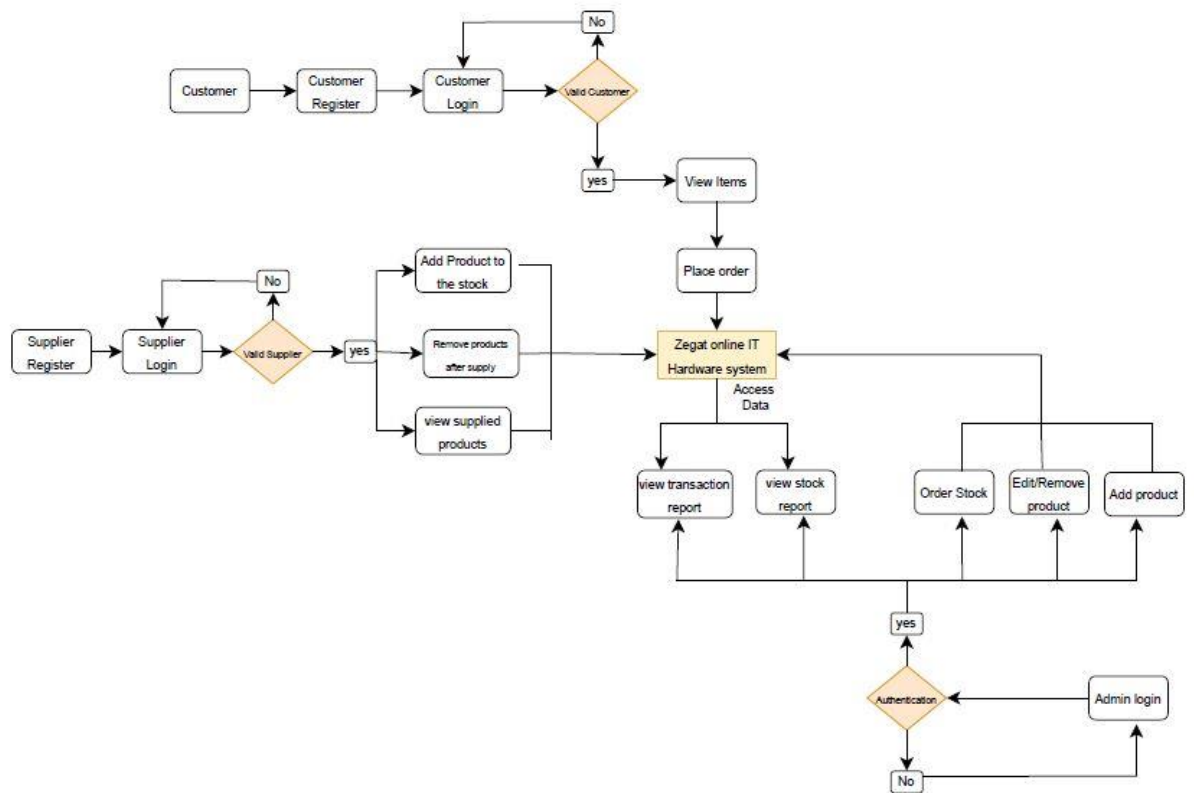
The project aims to create the development of a Zegat Online IT-Hardware Ordering application in which admin, supplier, and customer are the main modules. Where Admin maintains supplier and customer. The product can be managed by the supplier and admin and the customer who acts as an interface, can order the required product and go to the transaction to pay the bill.

2.4 Validations

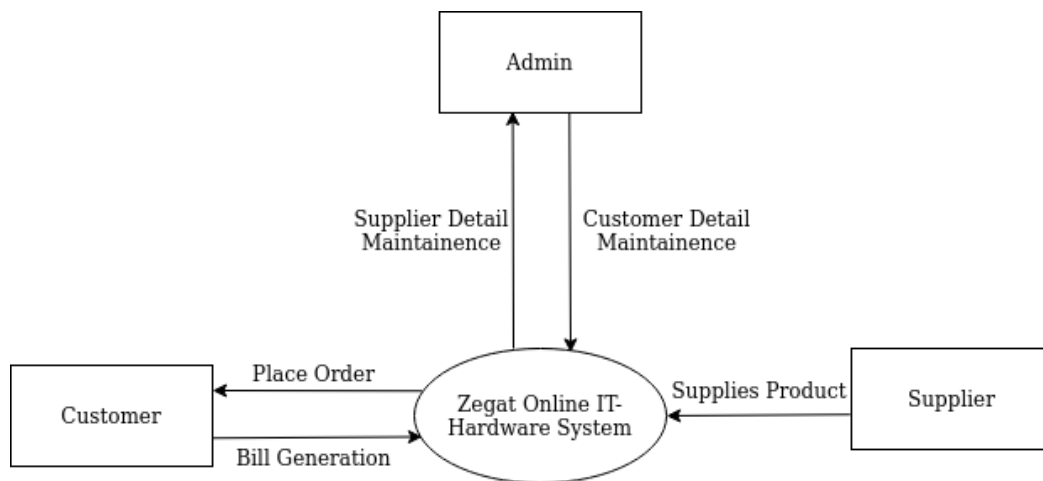
- In Zegat Online IT-Hardware Ordering System main menu, the choice entered must be digits only.
- The customer_name should be alphabet only and should be between 5-25 characters.
- While entering the customer_address, the address should not start with space and should be between 4-40 characters only.
- Mobile_number should be in digits only with the range of 10.
- The product id and quantity of the product should be in digits only.
- The names of the customer, supplier, admin, and product should be in alphabet only.
- Number of units and cost of the product should be in digits.
- Address should not be blank.
- Admin can log in only if the password is validated.

3. Detailed System design

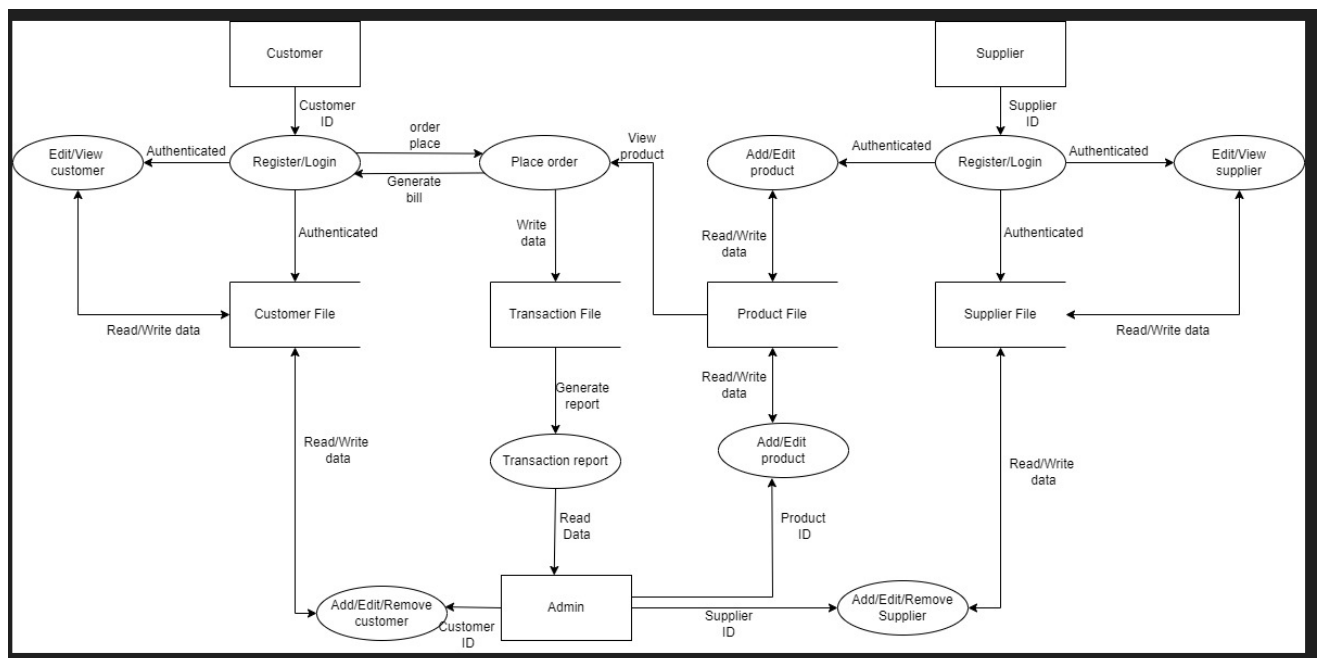
3.1 Flow chart



3.2 DFD of Level 0

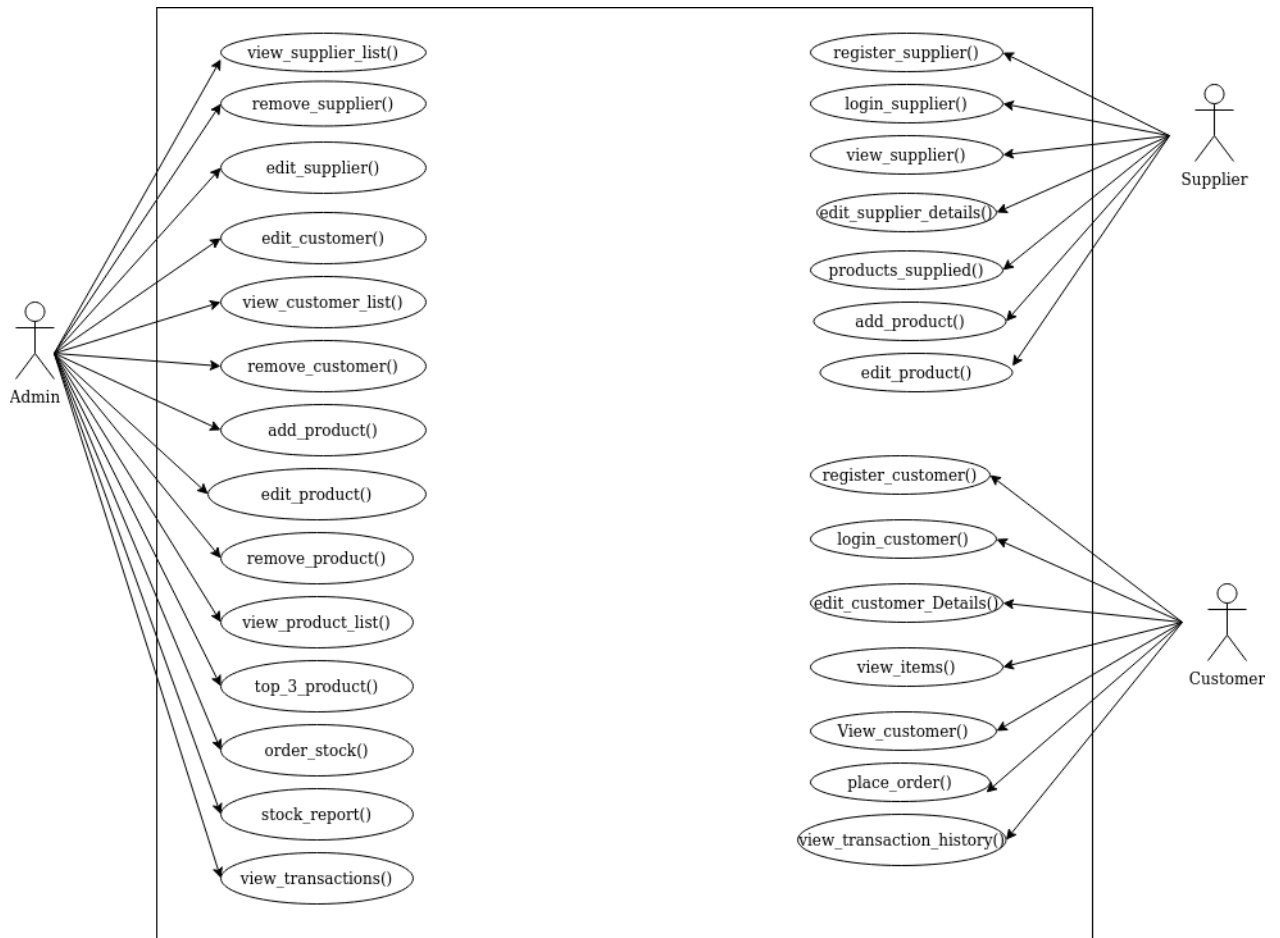


3.3 DFD of Level 1

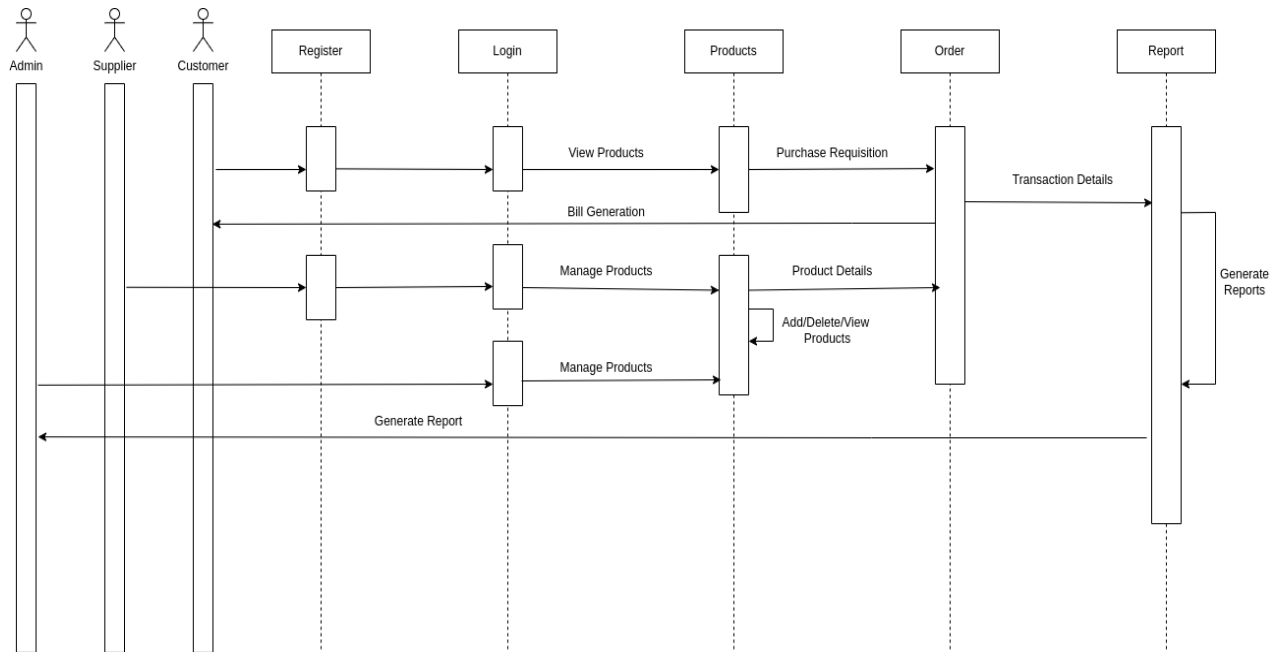


3. UML Diagrams

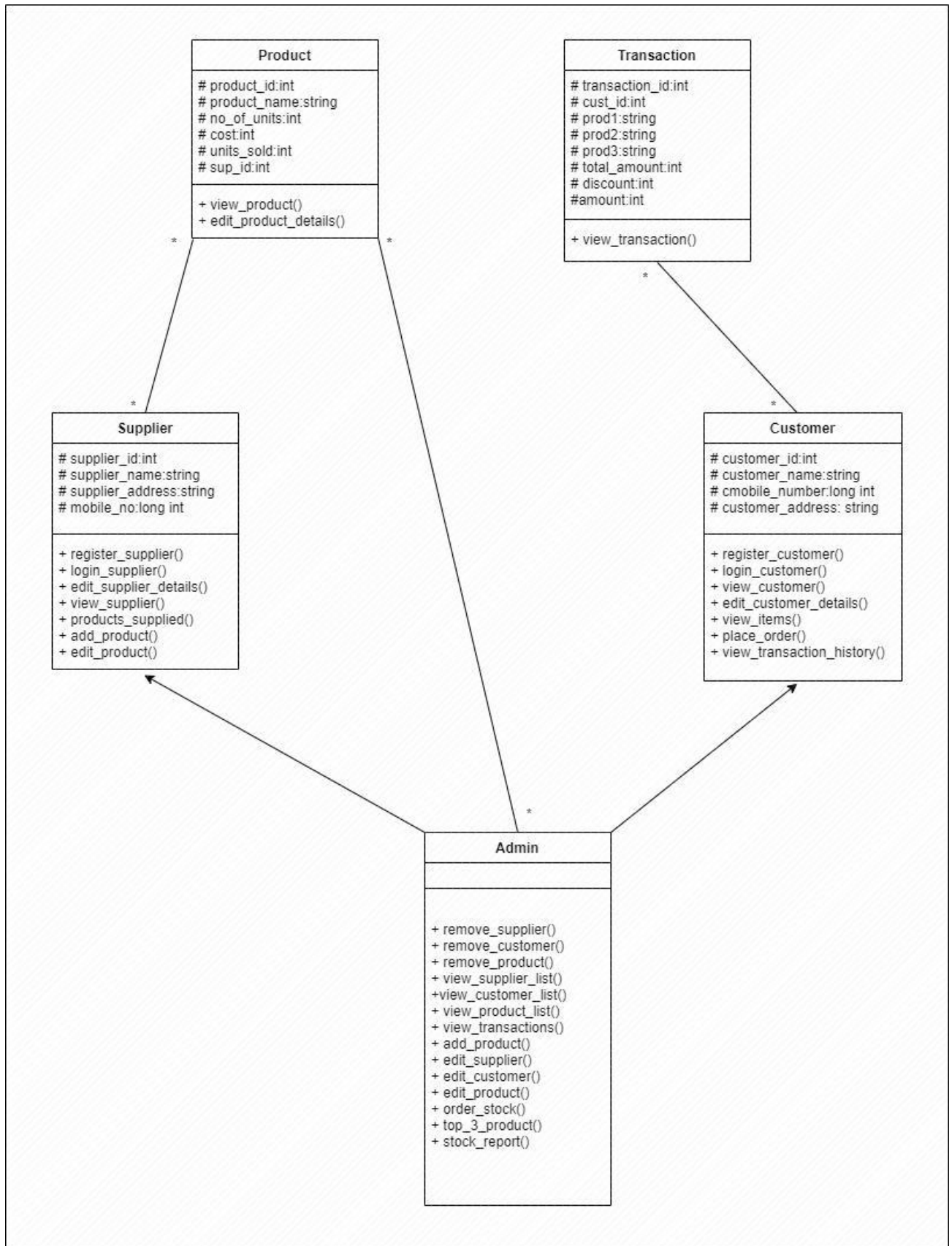
3.1 Use Case Diagram



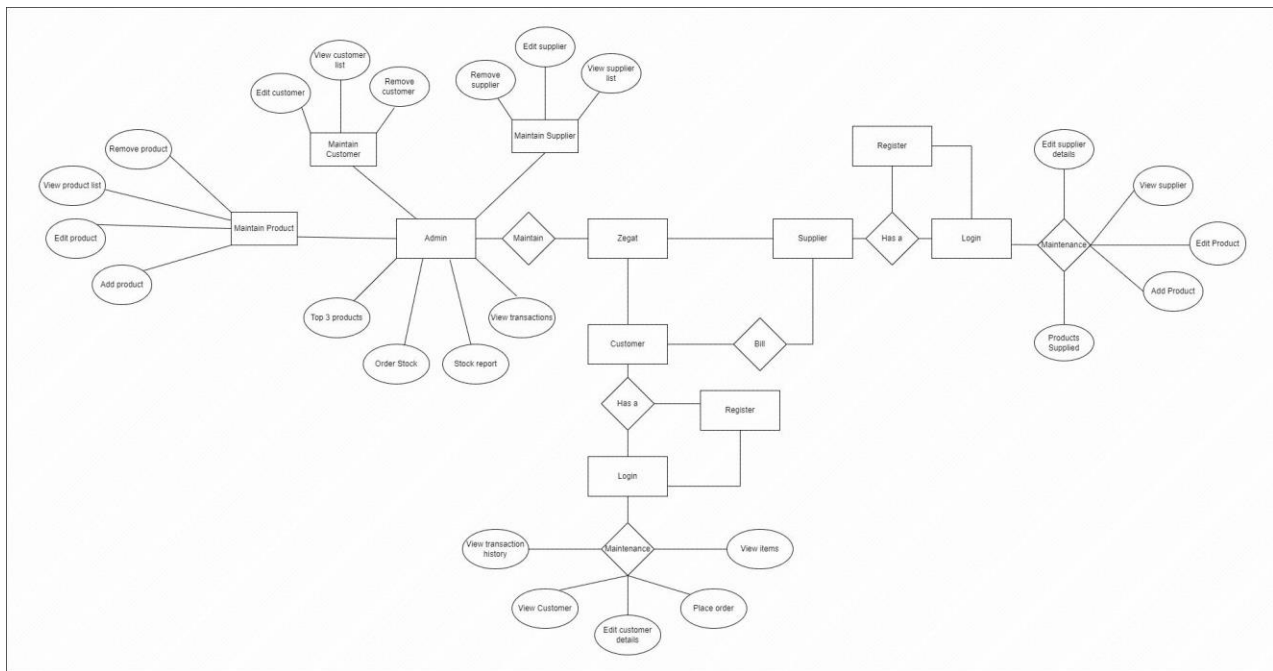
3.2 Sequence Diagram



4.3 Class Diagram



4.4 ER Diagram



5. Environment Description

5.1 Time zone support - IST

5.2 Language Support - English

5.3 User Desktop Requirements-

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

5.4 Server-side requirements

- 32 64-bit processors, 1 GHz or faster
- At least 1 GB free hard drive space
- At least 1GB RAM

5.4.1. Deployment Considerations: -

- Easy setup: no session storage daemon, use tmpfs and memory caching to enhance performance.
- Local storage is used

- No network latency to consider
- To scale buy a bigger CPU, more memory, larger hard drive, or additional hardware

Application Server Disk Space: -

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

5.4.2 Database Server Disk Space: -

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

5.4.3 Integration Requirements: -

- a. Language: - C++
- b. Tools: - Gcov, Valgrind, Makefile, Cpp unit, CPP check, Gprof
- c. Compiler: - g++
- d. Linux Environment

5.6 Configuration

5.6.1 Operating System - Linux Environment

6.Experimental Results:

6.1 Unit testing:

- Unit Testing with error

```
cguser4@instance-1: ~/Online_IT_Store_Project/cut/code/src$ ./cpptest
test::test_register_supplier
Enter Details for Supplier Registration
-----
Enter Supplier Name: Rahul Gupta
Enter Supplier address: Mumbai, Maharashtra
Enter Supplier Mobile number: 0987654321

Supplier Added Successfully!!!
Supplier Id is 101

: assertion
test::test_edit_supplier_details
Enter Details for Editing Supplier
-----
Enter Supplier Name: Rahul Gupta
Enter Supplier address: Mumbai, Maharashtra
Enter Supplier Mobile number: 9087654321

Details edited Successfully!!!

: assertion
test::test_register_customer
Enter Details for Customer Registration
-----
Enter Customer Name: Rajesh Kumar
Enter Customer address: Pune
Enter Customer Mobile number: 9870654321

Customer Added Successfully!!!
Customer Id is 1001
```

```
cguser4@instance-1: ~/Online_IT_Store_Project/cut/code/src$ ./cpptest

: assertion
test::test_register_customer
Enter Details for Customer Registration
-----
Enter Customer Name: Rajesh Kumar
Enter Customer address: Pune
Enter Customer Mobile number: 9870654321

Customer Added Successfully!!!
Customer Id is 1001

: assertion
unit_test.cpp:46:Assertion
Test name: test::test_register_supplier
assertion failed
- Expression: l==myobj1->register_supplier()

unit_test.cpp:51:Assertion
Test name: test::test_edit_supplier_details
assertion failed
- Expression: l==myobj1->edit_supplier_details()

unit_test.cpp:56:Assertion
Test name: test::test_register_customer
assertion failed
- Expression: l==myobj2->register_customer()

Failures !!!
Run: 3 Failure total: 3 Failures: 3 Errors: 0
cguser4@instance-1:~/Online_IT_Store_Project/cut/code/src$
```

- Unit testing without error

```

cguser4@instance-1:~/Online_IT_Store_Project/cut/code/src$ ./cpptest
test::test_register_supplier
Enter Details for Supplier Registration
-----
Enter Supplier Name: Rahul Gupta
Enter Supplier address: Mumbai, Maharashtra
Enter Supplier Mobile number: 1234567890

Supplier Added Successfully!!!
Supplier Id is 101

: OK
test::test_edit_supplier_details
Enter Details for Editing Supplier
-----
Enter Supplier Name: 9087654321
Name should be alphabets only and should be between 5-25 characters
Enter Supplier Name: Rahul Gupta
Enter Supplier address: Mumbai, Maharashtra
Enter Supplier Mobile number: 9087654321

Details edited Successfully!!!

: OK
test::test_register_customer
Enter Details for Customer Registration
-----
Enter Customer Name: Rajesh
Enter Customer address: Pune, Mumbai
Enter Customer Mobile number: 9078563421

```

```

cguser4@instance-1: ~/Online_IT_Store_Project/cut/code/src

Supplier Added Successfully!!!
Supplier Id is 101

: OK
test::test_edit_supplier_details
Enter Details for Editing Supplier
-----
Enter Supplier Name: 9087654321
Name should be alphabets only and should be between 5-25 characters
Enter Supplier Name: Rahul Gupta
Enter Supplier address: Mumbai, Maharashtra
Enter Supplier Mobile number: 9087654321

Details edited Successfully!!!

: OK
test::test_register_customer
Enter Details for Customer Registration
-----
Enter Customer Name: Rajesh
Enter Customer address: Pune, Mumbai
Enter Customer Mobile number: 9078563421

Customer Added Successfully!!!
Customer Id is 1001

: OK
OK (3)
cguser4@instance-1:~/Online_IT_Store_Project/cut/code/src$

```

6.2 Integration testing

- IT CASE 1: Supplier Account Registration

```

Enter Details for Supplier Registration
-----
Enter Supplier Name: twinkle
Enter Supplier address: mumbai maharashtra 400605
Enter Supplier Mobile number: 9876543210

Supplier Added Successfully!!!
Supplier Id is 101

-----
                        Supplier Corner
-----

        1.Supplier Registration
        2.login
        3.Exit

Enter your choice:


```

- IT_CASE 2: View Details of Supplier

```

Showing Details of Supplier
-----
        Supplier Id: 101
        Supplier Name: twinkle
        Supplier Address: mumbai maharashtra 400605
        Supplier Mobile: 9876543210

-----
                        Supplier DashBoard
-----
        Welcome, Supplier: twinkle

        1.View Details
        2.Edit Details
        3.Add Product
        4.Edit Supplied Product
        5.View Product Supplied
        6.Exit

Enter Choice:


```

- IT_CASE 3: Adding Product By Supplier

```

Enter Details of Product
-----
Enter Product Name: jjj

Name should be more than 5 and less than 20 character
Enter Product Name: sanfroz
Enter No of units: 45
Enter Cost of each item: 125
Cost cannot be less than RS.500!!!
Enter Cost of each item: 600

Product Added Successfully!!!
Product Id is 9001

```

- IT_CASE 4: Customer Account Registration

```

Enter Details for Customer Registration
-----
Enter Customer Name: rajababu
Enter Customer address: rajsthan alwara tijara 500145
Enter Customer Mobile number:
987654321
Mobile number should be 10 digits
Enter Customer Mobile number: 1234567890

Customer Added Successfully!!!
Customer Id is 1001

-----
                        Customer Corner
-----

    1.Customer Registration
    2.login
    3.Exit

Enter your choice:
☐

```

- IT_CASE 5: Place Order by Customer


```

Enter Product Id to purchase: 9002
Enter to Quantity: 1
Press 0 to Stop Shopping or any other to continue: 1

Enter Product Id to purchase: 9006
Sorry Product with 9006 id not found!!!

Press 0 to Stop Shopping or any other to continue: 1

Enter Product Id to purchase: 9001
Enter to Quantity: 4
Press 0 to Stop Shopping or any other to continue: 0

Showing Bill Details
-----
Transaction_id: 3001
Customer Id: 1002
Product 1: xiaomi, Quantity: 1
Product 2: samsung, Quantity: 4
Product 3: Nil
Total_amount: 9000
Discount: 900
Amount: 8100

Thank You For Visiting Our Shop!!!

```

- IT_CASE 6: Viewing list by Admin

```

-----Viewing Product List-----
ID      Name      Units      Cost      Supplied_By
9002     nokia       900       12345     101
9003     redmi       560       7894      101
9004     xiaom       4567      1234      101

-----
                          Maintain Product
-----

1.Add Product
2.Edit Product
3.Remove Product
4.View List of Product
5.Exit

Enter your choice:


```

- IT_CASE 7: View Transaction History by Customer

Transaction History								
Id	Customer_ID	Product_1	Product_2	Product_3	Total_amount	Discount	Amount_Paid	
3001	1002	xiaomi, Quantity: 1	samsung, Quantity: 4	Nil	9000	900	8100	

- IT_CASE 8: Stock Report

Product_Id	Name	Units_Available	Units_Sold	Price
9001	samsung	452	4	1000
9002	xiaomi	788	1	5000
9003	redmi	963	0	1234

7. Tools Report

7.1 Gprof

```

sgosw@instance-1: ~/Online_IT_Store_Project/sub/tech_report/gprof
[33] 0.0 0.00 0.00 4/6 std::_cxx11::list<transaction, std::allocator<transaction> >::end() [43]
std::_List_iterator<transaction>::_List_iterator(std::_detail::_List_node_base*) [33]
0.00 0.00 1/6 customer::place_order() [124]
0.00 0.00 1/6 supplier::add_product() [127]
0.00 0.00 2/6 Admin::top_three_product() [119]
0.00 0.00 2/6 product_list_to_file() [103]
[34] 0.0 0.00 0.00 6 std::operator!=(std::_List_iterator<product> const&, std::_List_iterator<product> const&) [34]
0.00 0.00 1/5 supplier::login_supplier() [128]
0.00 0.00 1/5 supplier::register_supplier() [130]
0.00 0.00 1/5 std::_cxx11::list<supplier, std::allocator<supplier> >::push_back(supplier const&) [
278] 0.00 0.00 2/5 supplier_list_to_file() [107]
[35] 0.0 0.00 0.00 5 std::_cxx11::list<supplier, std::allocator<supplier> >::end() [35]
0.00 0.00 5/8 std::_List_iterator<supplier>::_List_iterator(std::_detail::_List_node_base*) [19]
0.00 0.00 2/4 supplier::login_supplier() [128]
0.00 0.00 2/4 customer::login_customer() [125]
[36] 0.0 0.00 0.00 4 validations::take_choice() [126]
0.00 0.00 4/14 validations::check_digit(std::_cxx11::basic_string<char, std::char_traits<char>, std
::allocator<char> >) [13]
0.00 0.00 4/23 std::_cxx11::stoi(std::_cxx11::basic_string<char, std::char_traits<char>, std::allo
cator<char> > const&, unsigned long*, int) [9]
0.00 0.00 1/4 supplier::corner() [101]
0.00 0.00 1/4 supplier::register_supplier() [130]
0.00 0.00 1/4 supplier::file_to_list() [106]
0.00 0.00 1/4 Admin::Admin() [120]
[37] 0.0 0.00 0.00 4 supplier::supplier() [137]
0.00 0.00 1/4 supplier::corner() [101]
0.00 0.00 1/4 supplier::file_to_list() [106]
0.00 0.00 1/4 void __gnu_cxx::new_allocator<std::_List_node<supplier> >::destroy<supplier>(supplier
&) [151]
0.00 0.00 1/4 Admin::~Admin() [319]
[38] 0.0 0.00 0.00 4 supplier::~supplier() [38]

```

7.2 Gcov

```

-: 0:Source:func1.cpp
-: 0:Graph:main.gcno
-: 0:Data:main.gcda
-: 0:Runs:1
-: 1:/*****
-: 2:
-: 3:
-: 4:FILENAME: func1.cpp
-: 5:DESCRIPTION:This file is used to display the main menu to the user.
-: 6:
-: 7:DATE          NAME          REASON
-: 8:-----
-: 9:22/9/22      Group-4        This is done for creation of menu
-: 10:
-: 11: *****/
-: 12:#ifndef FUNC1
-: 13:#define FUNC1
-: 14:#include <iostream>
-: 15:#include <string>      //Includes string functions
-: 16:#include <iomanip>
-: 17:#include <limits>
-: 18:#include "header.hpp"
-: 19:#include "validations.cpp"
-: 20:#include <mutex>
-: 21:#include <thread>
-: 22:mutex accum_mutex;
-: 23:
-: 24:/*****
-: 25:
-: 26:
-: 27:FUNCTIONNAME: register_customer()
-: 28:DESCRIPTION:This function is used to register customer.
-: 29:
-: 30: *****/
function_ZN8customer17register_customerEv called 0 returned 0% blocks executed 0%
#####: 31:int customer::register_customer()
-: 32:{
#####: 33:    int id=0;
#####: 34:    if(cust.empty()) //Checting if customer list is empty.
call    0 never executed
branch  1 never executed
branch  2 never executed
-: 35:    {
#####: 36:        id=1000;
-: 37:    }
#####: 38:    for(customer c:cust) //Iterating through customer list.
call    0 never executed
call    1 never executed
call    2 never executed
branch  3 never executed
branch  4 never executed
call    5 never executed
call    6 never executed
branch  7 never executed
branch  8 never executed

```

7.3 Cppcheck

```
cguser4@instance-1: ~/Online_IT_Store_Project/cut/tools_report/cppcheck
cguser4@instance-1:~/Online_IT_Store_Project/cut/tools_report/cppcheck$ cat C4_cppcheck_all.txt
menu.cpp:45:3: style: Exception should be caught by reference. [catchExceptionByValue]
  catch(exception ex)
  ^
menu.cpp:100:3: style: Exception should be caught by reference. [catchExceptionByValue]
  catch(exception ex)
  ^
menu.cpp:152:3: style: Exception should be caught by reference. [catchExceptionByValue]
  catch(exception ex)
  ^
menu.cpp:206:3: style: Exception should be caught by reference. [catchExceptionByValue]
  catch(exception ex)
  ^
menu.cpp:269:3: style: Exception should be caught by reference. [catchExceptionByValue]
  catch(exception ex)
  ^
menu.cpp:319:3: style: Exception should be caught by reference. [catchExceptionByValue]
  catch(exception ex)
  ^
main.cpp:46:5: style: Exception should be caught by reference. [catchExceptionByValue]
  catch(exception ex)
  ^
menu.cpp:404:9: style: The comparison 'flag == 0' is always true. [knownConditionTrueFalse]
  if(flag==0 as choice!=6)
  ^
menu.cpp:354:11: note: 'flag' is assigned value '0' here.
  int flag=0;
  ^
menu.cpp:404:9: note: The comparison 'flag == 0' is always true.
  if(flag==0 as choice!=6)
  ^
menu.cpp:477:9: style: The comparison 'flag == 0' is always true. [knownConditionTrueFalse]
  if(flag==0 as choice!=6)
  ^
menu.cpp:423:11: note: 'flag' is assigned value '0' here.
  int flag=0;
  ^
menu.cpp:477:9: note: The comparison 'flag == 0' is always true.
  if(flag==0 as choice!=6)
  ^
```

7.4 Valgrind

```
cguser4@instance-1: ~/Online_IT_Store_Project/cut/tools_report/valgrind
cguser4@instance-1:~/Online_IT_Store_Project/cut/tools_report/valgrind$ cat valgrind_without_error_group4
==207191== Memcheck, a memory error detector
==207191== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==207191== Using Valgrind-3.16.1 and LibVEX; rerun with -h for copyright info
==207191== Command: test
==207191==
==207191==
==207191== HEAP SUMMARY:
==207191==    in use at exit: 0 bytes in 0 blocks
==207191==   total heap usage: 196 allocs, 196 frees, 10,215 bytes allocated
==207191==
==207191== All heap blocks were freed -- no leaks are possible
==207191==
==207191== For lists of detected and suppressed errors, rerun with: -s
==207191== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
cguser4@instance-1:~/Online_IT_Store_Project/cut/tools_report/valgrind$
```

8. Reference:

- Prof. Prakash Biswa(Mentor)
- <https://www.w3schools.com/cpp/>
- <https://www.geeksforgeeks.org/c-plus-plus/>
- <https://www.tutorialspoint.com/cplusplus/index.htm>
- <https://www.geeksforgeeks.org/list-cpp-stl/>