VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology& Engineering

M.Tech Software Engineering

PROJECT REPORT

SWE2019 – DESIGN PATTERNS

Slot: A1

IMPROVING THE DESIGN OF CLOTH SHOP MANAGEMENT SYSTEM USING DESIGN PATTERNS

SUBMITTED BY

16MIS0102-K.KARTHIKEYAN

FACULTY INCHARGE

Prof SENTHIL KUMARAN.U

PROBLEM STATEMENT

Consider a application like cloth shop management Software For Designing a system is perhaps the most critical factor affecting the quality of the software. No design methodology reduces the process of design to a sequence of mechanical steps. It only provides guidelines to aid the designer during the design process. Fortunately, and other researchers proposed many design patterns that give us solution of general design problems. Design decisions of experienced designers are recorded in form of design patterns. Thus each design pattern focuses on a particular object-oriented design problem or issue. If these design patterns are used by a novice designer, the obtained system design would be much better. In this paper, i designed a very simple cloth shop management system using object oriented approach without design patterns. All the defects are shown and this design is improved using few design patterns. I used three design patterns: **singleton** and **prototype** Design Patterns in this study.

INTRODUCTION:

Demand of software has been increasing day by day. Therefore, there is more demand for software development paradigm that improves quality and productivity of software development.

Problem Statement and Analysis: I have taken a very simplified version of cloth shop management system that manages the category of shop maintainence. It performs functions such as managing item and order. It also include function that enable users to search their resources in shop. To analyze and understand the system under study i will develop a model using object-oriented approach. Whenever order class requests for a item, item class instantiates an instance of item, issuing it to the requesting order.

A design pattern isn't a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations. They guarantee the creation of transparent structures which allow software to be easily understood, extended, and reused. The description of design patterns provides information about the structure, the participant's roles, the interaction between participants and, above all, the intent for which they should be used. In this paper, i will be implementing design patterns in cloth shop management system to improve its design that is obtained without using design patterns. I choose the cloth shop management system as a domain because of its simplicity. Further i designed the system for few requirements to show the objective of use of design patterns in design for improvement. In the next section, i will explain the system under study- Cloth Management System, its design without using design patterns and defects in this design. In the next section, i will describe two design patterns: singleton and prototype design patterns that are used in this work to improve the design.

CLOTH SHOP MANAGEMENT SYSTEM:

The following has been designed with a deep insight into the working of cloth shop management system. The following project can be used to manage products, order details. The product ID and Bill no are assigned through the project to prevent loss of information. It includes options such as modifying and deleting records to update the product and order database. Product and order details can be searched when required. The project also generates the report for bill charged. While taking any input from the user all possible validation checks have been taken care of to increase the robustness of the project. The source code of the project is developed by C# language in visual studio. It maintains stock list and provides bill for purchased items in shop. Sales and stock management is done here.

PROPOSED SYSTEM:

SINGLETON:

First the admin login class is considered as singleton and get method is written in the class for static instance declared. A function named login() is used for singleton to access. It defines an instance operation that lets the admin access its unique instance. It is the only way to enter the system.

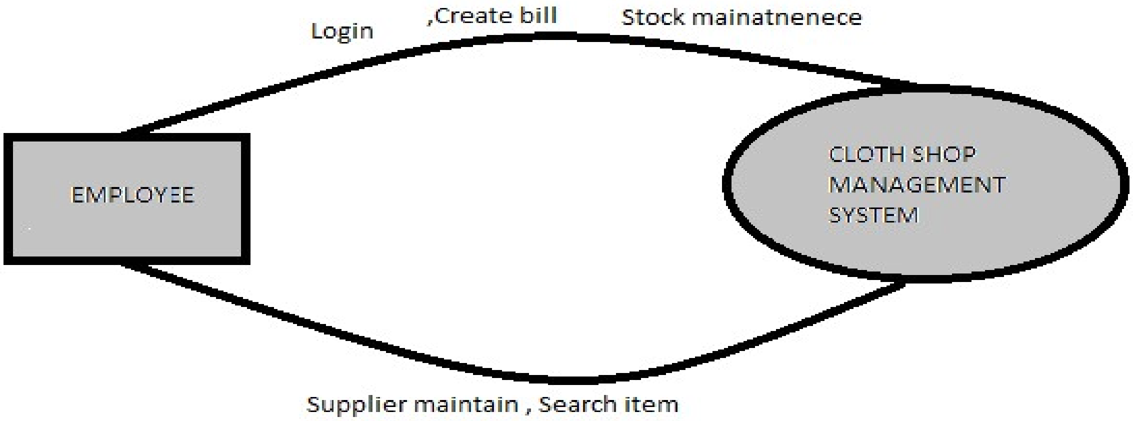
PROTOTYPE:

In this after the item and purchase data is stored, it is used for search and report to future reference. In this for order we need itemid, so we copy the details of item using itemid (price is taken from itemid). So when the itemid is selected and quantity is entered the calculate button is clicked. It gets the price of itemid and multiplies it with the quantity and delivers the total price of order. In this list class is created to get the item and order report.

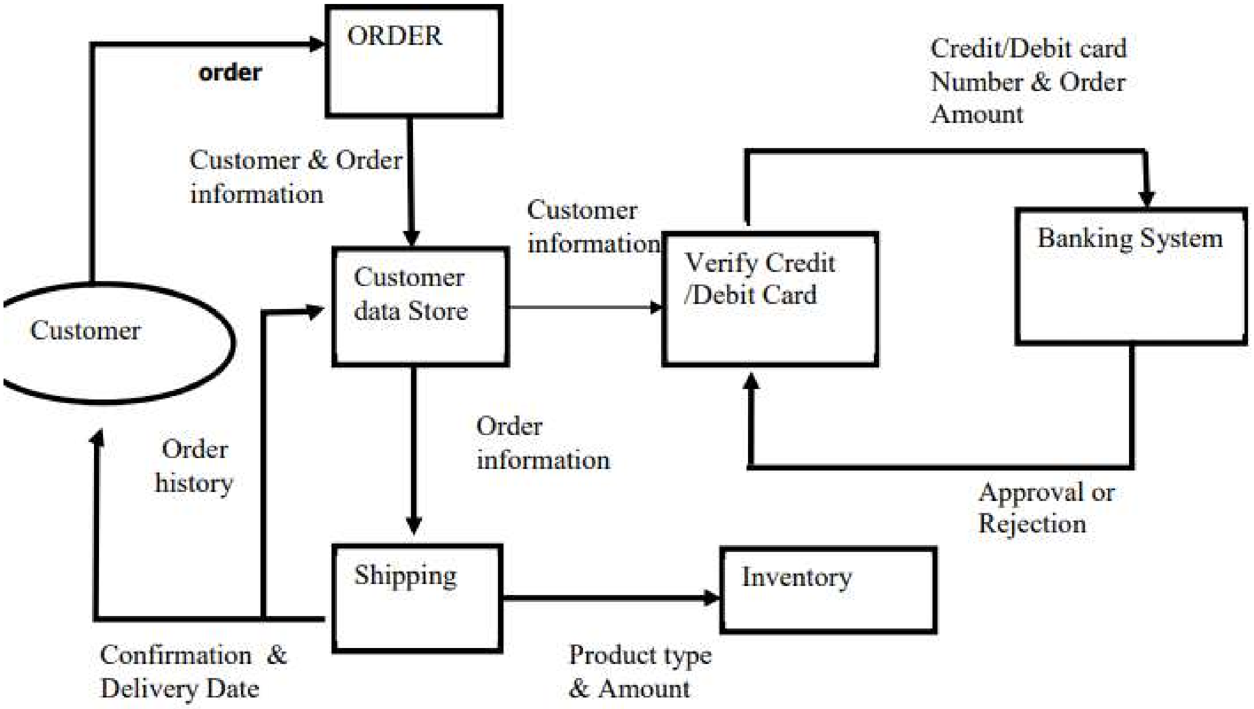
SYSTEM DESIGN:

ARCHITECTURE PATTERN:

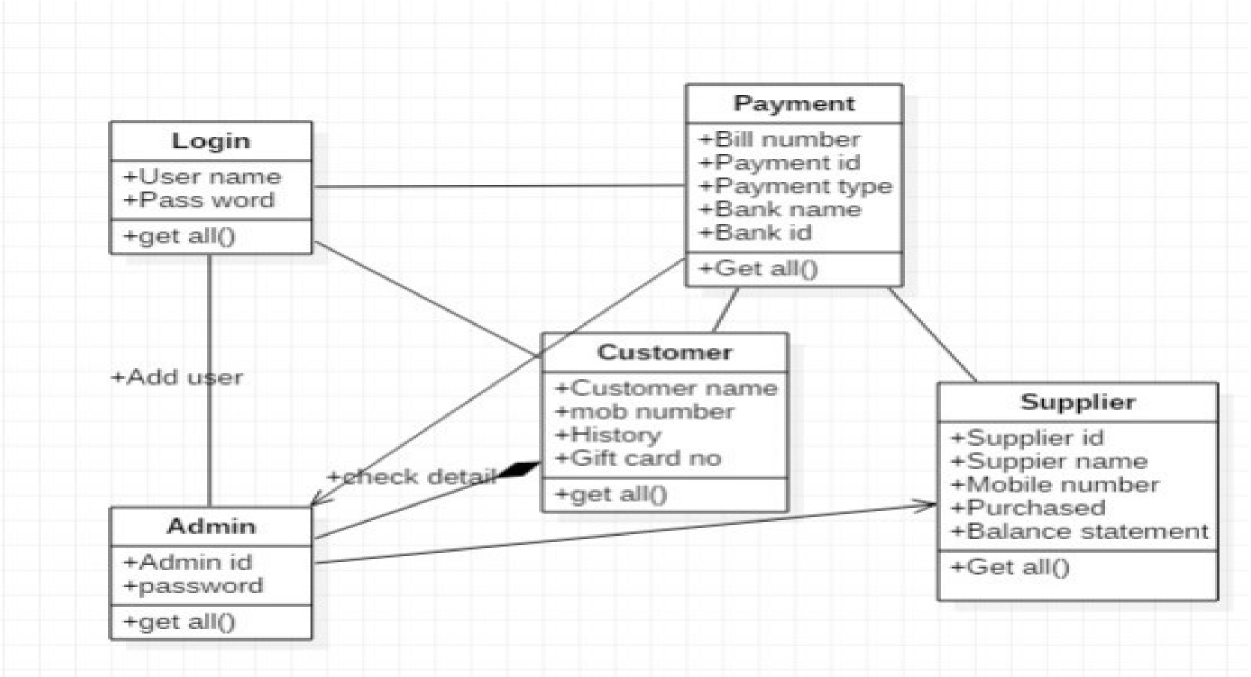
DFD-0



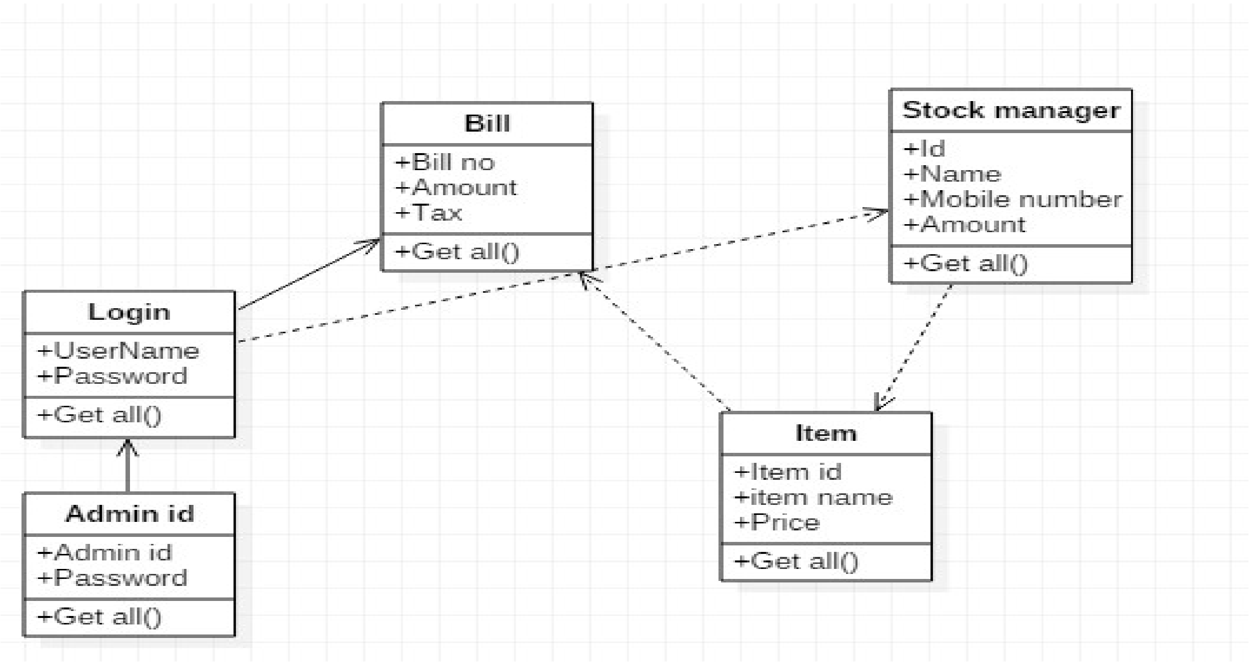
System design



MODIFIED SYSTEM DESIGN USING SINGLETON



MODIFIED SYSTEM DESIGN USING PROTOTYPE



HARDWARE REQUIREMENT:

Processor: Intel
RAM: 4GB
Hard disk: 10GB
Keyboard: PS2/USB
Mouse:PS2/USB

SOFTWARE REQUIREMENT:

Operating systems: windows 10

Front end: VB.net

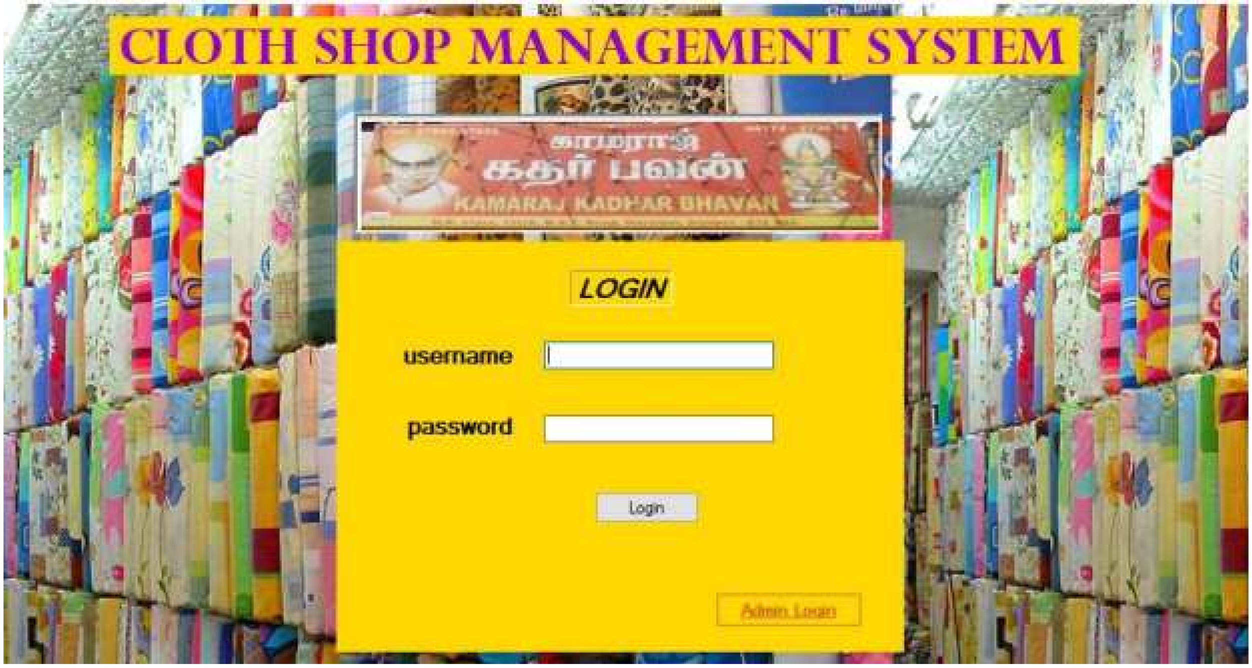
Back end: ADO.net

Software package: visual studio 2012

Output
Code for login using singleton

```
private void button1_Click(object sender, EventArgs e)
       {
           SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\dell\sourc
e\repos\WindowsFormsApp9\WindowsFormsApp9\test.mdf;Integrated
Security=True;Connect Timeout=30");
           SqlDataAdapter sda = new SqlDataAdapter("select count(*) from
login where username='" + textBox1.Text + "' and password='" +
textBox2.Text + "'", con);
           DataTable dt = new DataTable();
           sda.Fill(dt);
           if(dt.Rows[0][0].ToString()=="0")
           {

               this.Hide();
               Main2 mm = new Main2();
               mm.Show();
           }
           else
           {
               MessageBox.Show("Invalid username or password");
           }
       }
```

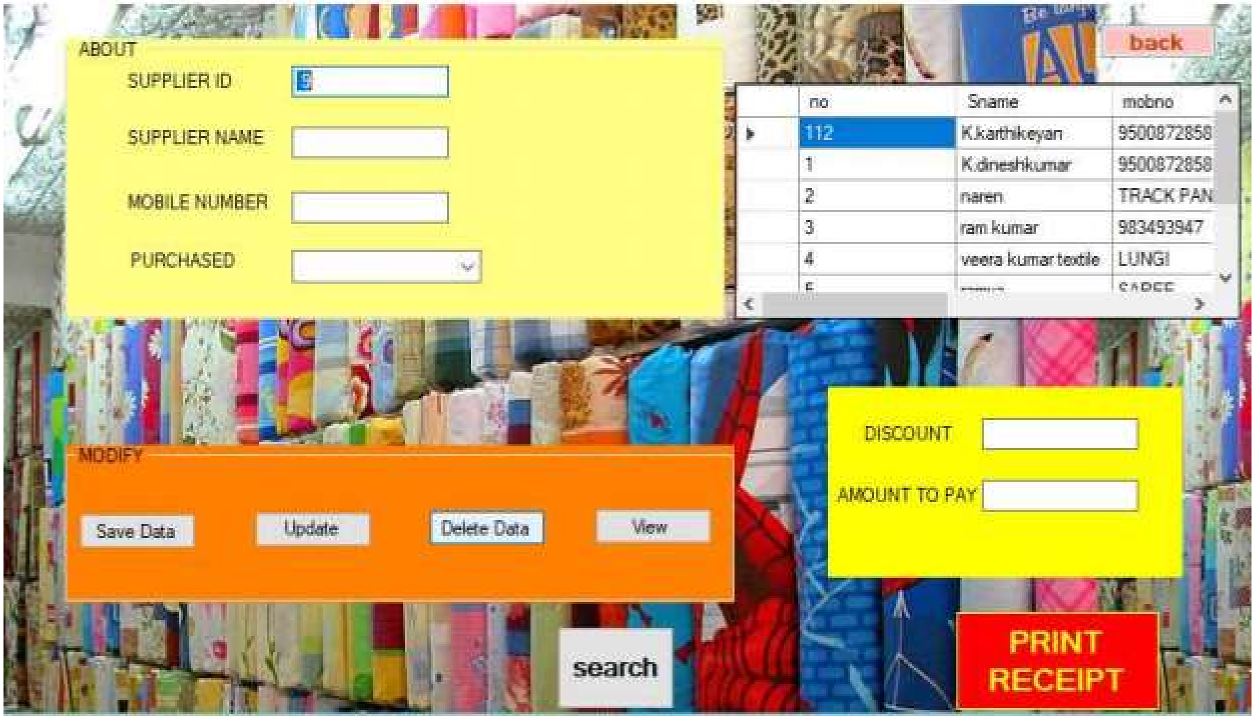
Output



Code using prototype

```
private void button1_Click(object sender, EventArgs e)
        {
            con.Open();

                int aa = Convert.ToInt32(textBox1.Text);

                string abc = "SELECT
no,Sname,amount,mobno,purchase,discount FROM tbl_insertion WHERE
no = '" + aa + "'";
                cmd = new SqlCommand(abc, con);

                MessageBox.Show("one found");


                dr = cmd.ExecuteReader();
                DataTable dt = new DataTable();
                dt.Load(dr);
                dataGridView1.DataSource = dt;
            con.Close();


        }
```

Output

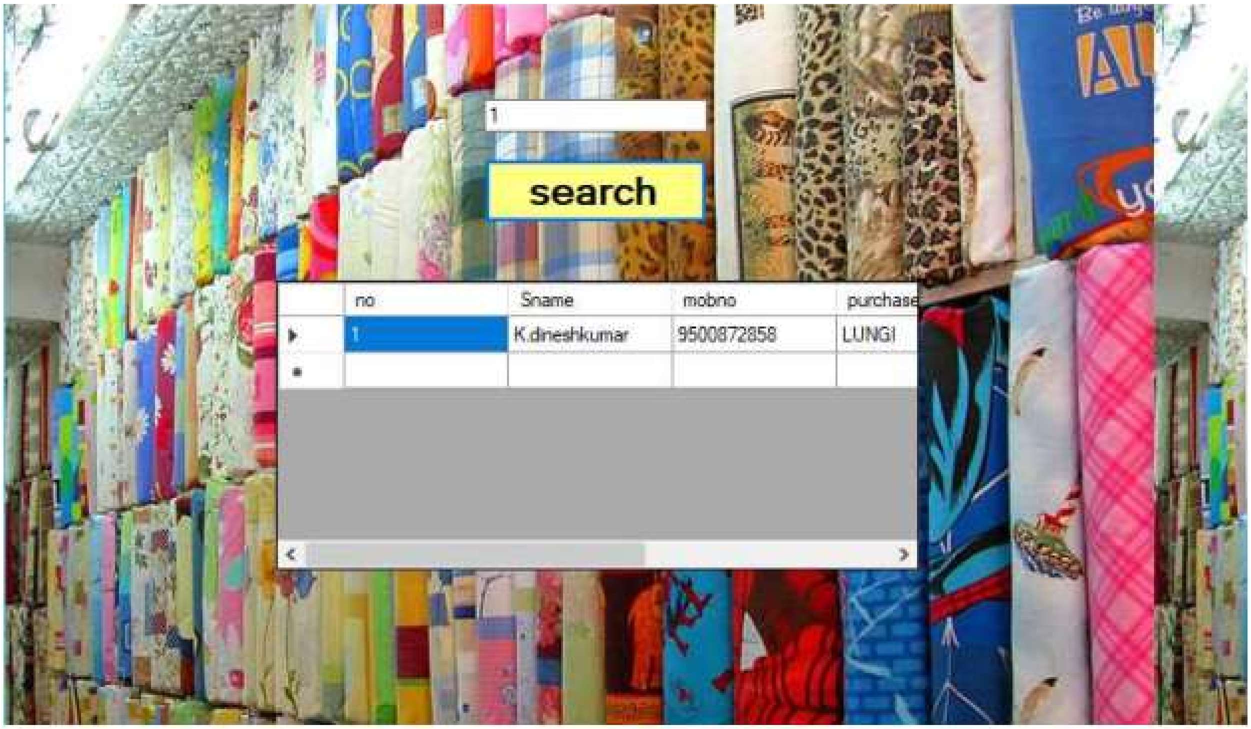


no	Sname	mobno
112	K.karthikeyan	9500872858
1	K.dineshkumar	9500872858
2	naren	TRACK PAN
3	ram kumar	983493947
4	veera kumar textile	LUNGI
5	[illegible]	SAREE

```
private void Main3_Load(object sender, EventArgs e)
        {
            this.loginTableAdapter.Fill(this.testDataSet.login);
            con.Open();
```

//Tbl insertion used as a prototype//

```
            SqlDataAdapter SDA = new SqlDataAdapter("select * from tbl_insertion", con);
            DataTable dt = new DataTable();
            SDA.Fill(dt);
            int i = dt.Rows.Count;
            string cnt = Convert.ToString(i + 1);
            textBox1.Text = " " + cnt;
            con.Close();

            using (testEntities db = new testEntities())
            {
                tblinsertionBindingSource.DataSource = db.tbl_insertion.ToList();
            }
        }
```



1

	no	Sname	mobno	purchase
▸	1	K.dineshkumar	9500872858	LUNGI
*				

CONCLUSION:

In this paper, we have proposed design of cloth shop management system with and without design patterns. We have shown few problems in design proposed without using design patterns. This design is then improved using two design patterns: SINGLETON and PROTOTYPE design patterns. We added singleton pattern login for unique access. It defines an instance operation that lets the admin access its unique instance.to responsibility of issuing details to shop system. Then, prototype pattern is implemented to get the details of shop system (item and order) and uses item id to get price and fetch common database, using which the user can be able to order the item.

REFRENCES:

1.[GHJV94] E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns (Elements of Reusable Object-Oriented Software); Addison Wesley, ISBN 0-201-63361-2 (1994)

2.[JSZ97a] J.-H. Jahnke, W. Schäfer, and A. Zündorf: Generic Fuzzy Reasoning Nets as a basis for reverse engineering relational database applications; to appear in: Proc. of ESEC'97 (to appear)

3.[Schürr91] A. Schürr: Operational Specification with PROgrammed GraphREwritingSystems; PhD Thesis (in German),RWTH Aachen 1991, Deutscher Universitäts Verlag, Vieweg, ISBN 3-8244-2021-X (1991)

4.[UML97a] The UML Notation Guide; Technical Report, Version 1.0, Rational Software, Santa Clara, http://www.rational. com (1997)

5.[Zündorf96] A. Zündorf: An Development Environment for PROgrammed GraphREwritingSystems - Specification, Implementation and Application; PhD Thesis (in German), RWTH Aachen 1995; Deutscher Universitäts Verlag, Vieweg, ISBN 3-8244-2075-9 (1996)

6. [FMW97] G. Florijn, M. Meijers,)