OBJECT ORIENTED ANALYSIS AND DESIGN FACTORY MANAGEMENT SYSTEM PROJECT

MEHMET HAMDİ USLU,2400000145

Abstract— This paper is about the project we were designated to prepare according to principals of object oriented design. We were given list of suggested systems and I chose Manufactoring system under a imaginary factory that produces and has multiple workers and sub-systems and I presented my project as a desktop application with swing.

1 INTRODUCTION

THIS document is to explain my project which theme is Manufactoring Systems. It is basicly simple databese for a factory. It has a graphical interface (GUI), for various of prefixed employement group which are manager, en-gineer, worker and other employees. Our program holds various menus for different employment groups. That means, managers enter with manager menu, engineers enter with engineer menu which those menus have own password. Other than that our program has an admin menu which has certain specifications. In this system employees can use system to manage products, produc-tion plans/specifications, orders and manufacturing machines. Managers can manage everything and employees for example to start up everything we need Administer which is us can add every group of people but mainly we need to add a manager to startup the system and let user carry on the administration.

2 REQUIREMENTS

Requirements are for this project are Encapsulation principle, examples of Polymorphism and Inheritance the usage of List, Stack or Queue data structures which you will be implemented which needed and must user. Users that are Manager, Engineer, Worker and other kind of employees. They all need to implemented using those principles on mind. Not every but most of them must at least represent single one of these needs.

3 DIAGRAM EXPLANATIONS

In my class diagram there is a Person class that is source for worker, engineer, managerand other employee. Persons have entering and leaving work times. Normal workers use machines and quality check, proucts. Engineers can organize view and add plans, fix malfunctions and view reports of the system. Managers also organize view and add plans and view reports of thesystem. Reports mentioned are Producement, Employee, System Staus, Malfunction and Vehicle Reports to monitor whole system. Other employees can also fix malfunctions however; they also use Vehicles to take or deliver orders to factory. Orders have two kinds that are Sale and Purchase orders. Finally, there is the Storage for producements and their needed functions.

Error diagram explains how malfunctions handled in the system. These are can only fixed by Worker or Engineer and if they cannot they summit their report with expertise help needed and solve the malfunction.

In state machine diagram we show Registered workers registers or comes to work and do what they are assigned to in Factory and worker in active state simulates leaving from work after shift.

Storage diagram handles necessary materials for factory it checks rate of production and decides how long raw material will last in current rate and if it needed imports fresh supply of material and also checks current fullness of the storage to create empty space if needed (deliver goods).

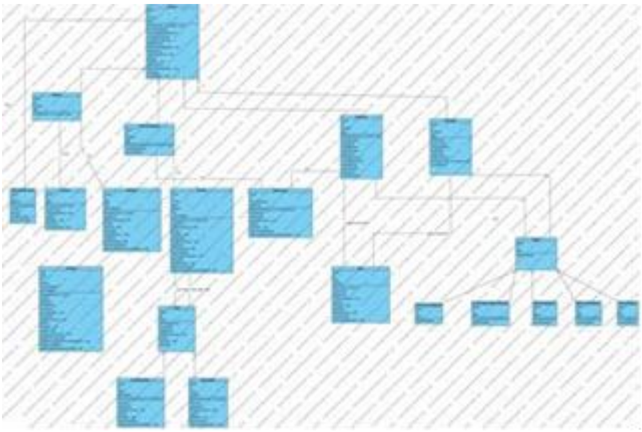


DIAGRAM 1: Class diagram

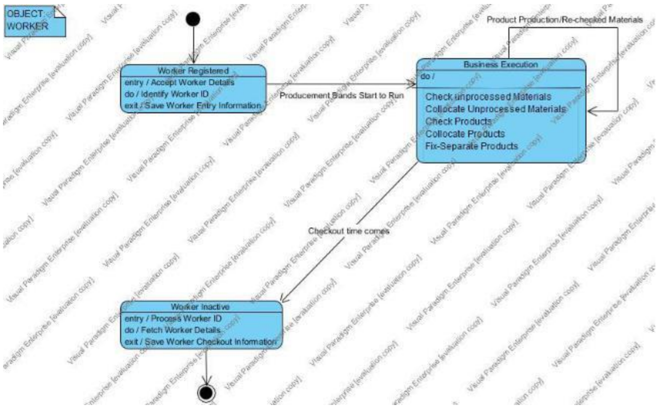

OBJECT: WORKER
Worker Registered
entry / Accept Worker Details
do / Identify Worker ID
exit / Save Worker Entry Information
Producement Bands Start to Run
Business Execution
do /
Check unprocessed Materials
Collocate Unprocessed Materials
Check Products
Collocate Products
Fix-Separate Products
Product Production/Re-checked Materials
Checkout time comes
Worker Inactive
entry / Process Worker ID
do / Fetch Worker Details
exit / Save Worker Checkout Information

DIAGRAM 2: Error diagram

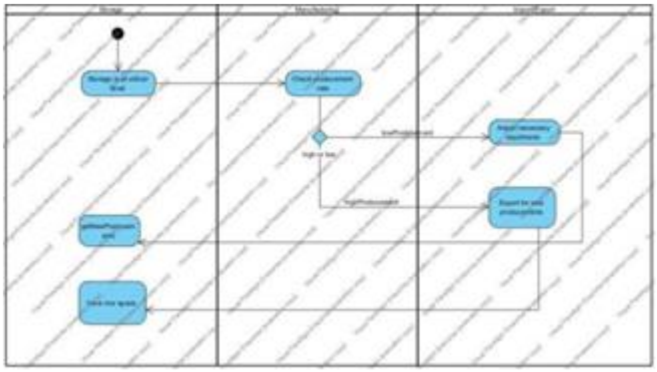


DIAGRAM 3: State diagram

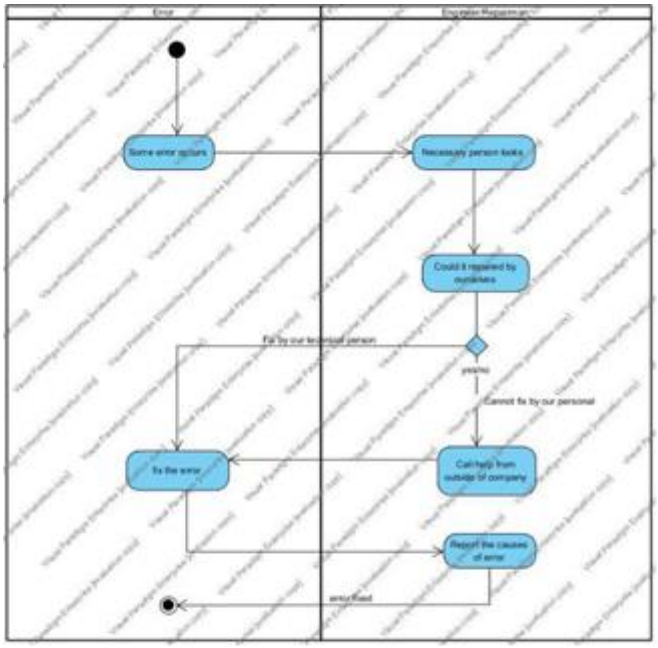
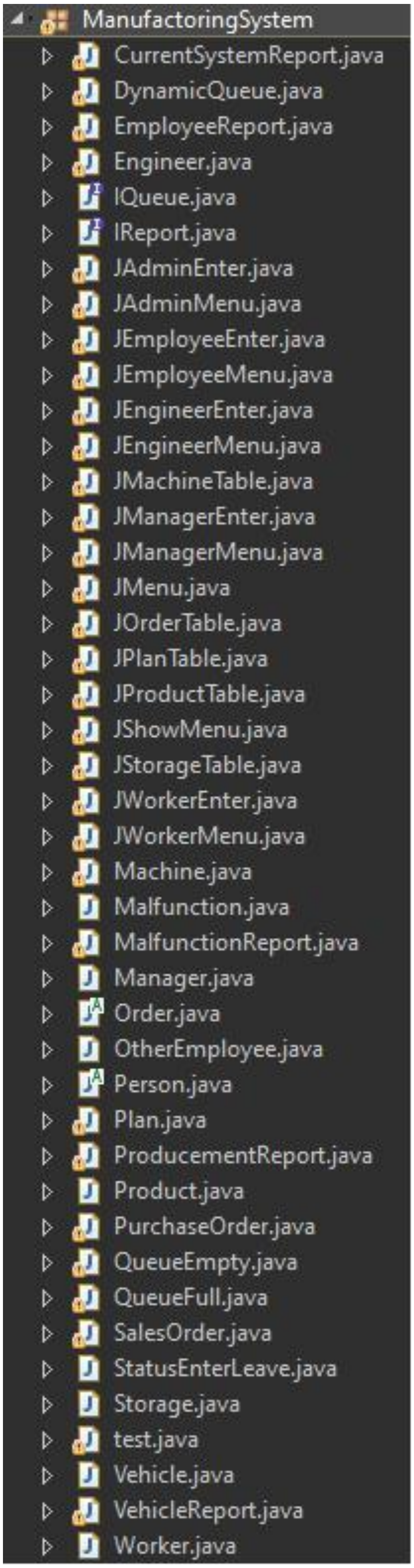

Error
Engineer Repairman
Some error occurs
Necessary person looks
Could it repaired by ourselves
Fix by our technical person
yes/no
Cannot fix by our personal
Fix the error
Call help from outside of company
Report the causes of error
error fixed

DIAGRAM 4: Storage diagram

4 Programming Everything

4.1 Opening Classes

First of all I needed to open my classes according to the diagram I drew which was quite easy to and create the variables which was declared on diagrams.



ManufactoringSystem
CurrentSystemReport.java
DynamicQueue.java
EmployeeReport.java
Engineer.java
IQueue.java
IReport.java
JAdminEnter.java
JAdminMenu.java
JEmployeeEnter.java
JEmployeeMenu.java
JEngineerEnter.java
JEngineerMenu.java
JMachineTable.java
JManagerEnter.java
JManagerMenu.java
JMenu.java
JOrderTable.java
JPlanTable.java
JProductTable.java
JShowMenu.java
JStorageTable.java
JWorkerEnter.java
JWorkerMenu.java
Machine.java
Malfunction.java
MalfunctionReport.java
Manager.java
Order.java
OtherEmployee.java
Person.java
Plan.java
ProducementReport.java
Product.java
PurchaseOrder.java
QueueEmpty.java
QueueFull.java
SalesOrder.java
StatusEnterLeave.java
Storage.java
test.java
Vehicle.java
VehicleReport.java
Worker.java

4.2 What is used where?

Interface and Abstract classes has been used and implemented on Person, Order, IReport. There is a main test class that is can work from eclipse console too yet there is also Swing UI. JFrames are crearted for every user type and other vital activities on factory like reports of the business practices, plans, orders and storage.

5 User credentials for system

FOR LOGINS:

Administor ID is : -1

Administor Password is : deuceng<3

Manager Ids are: 1000+id

Manager Password is : manager1234

Engineer IDs are : 2000+id

Engineer Password is : engineer1234

Worker IDs are : 3000+id

Worker Password is : worker1234

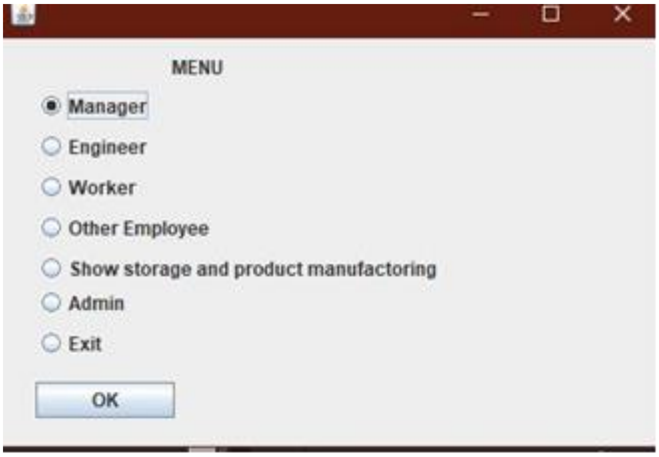
Employee IDs are : 4000+ id

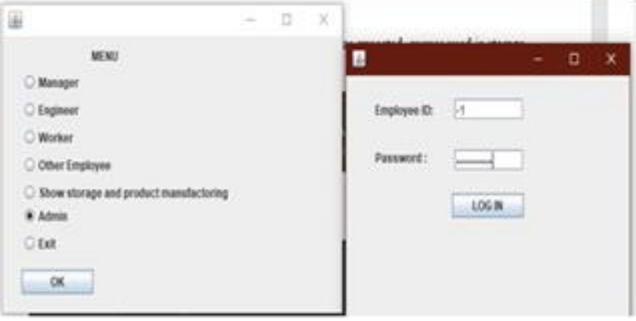
Employee Passwords is : employee1234

Storage Passwords is : storage1234

You can use 1000, 2000, 3000, 4000 without creating more users they are decleared for test purposes.

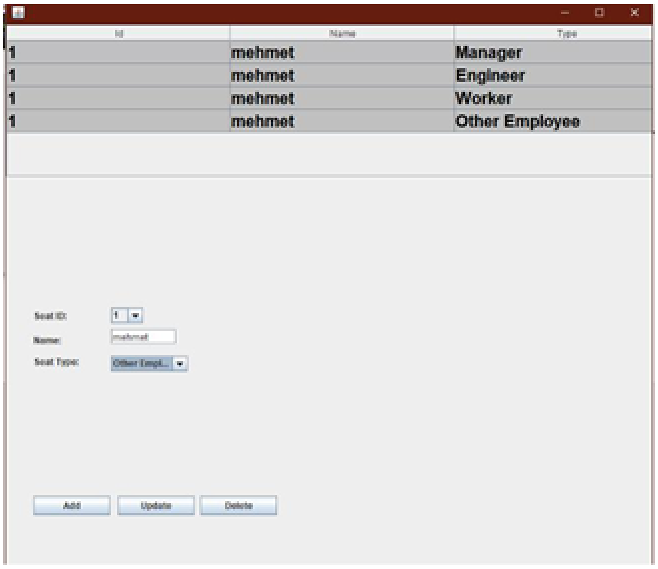
6 Screenshots





Employee ID: -1

Password :



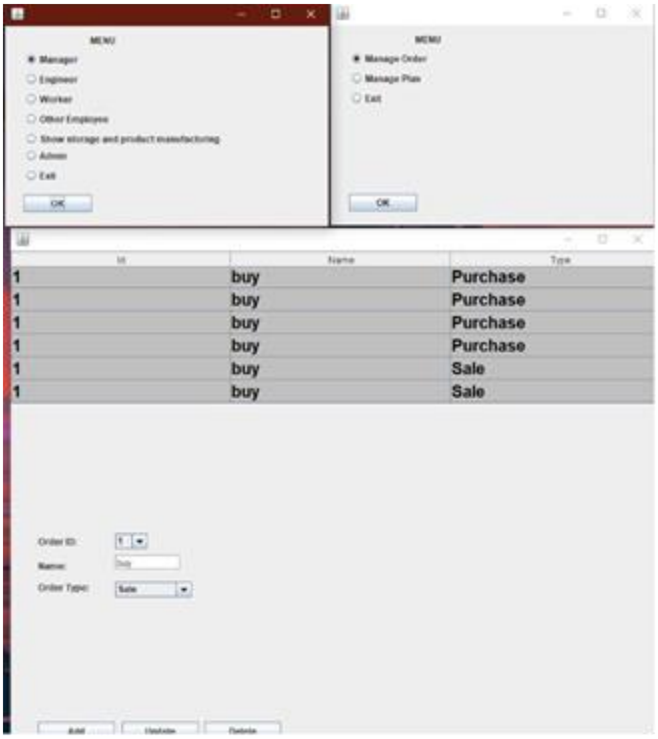
Id	Name	Type
1	mehmet	Manager
1	mehmet	Engineer
1	mehmet	Worker
1	mehmet	Other Employee

Seat ID: 1

Name: mehmet

Seat Type: Other Empl...

In my main menu. As decleared before each employee type has to enter with their employee type and their own ID and password. As an instance it is our admin menu after admin ID and password is correct. After successful login as can be seen everytype of worker can be added on admin menu and they can be updated.



Id	Name	Type
1	buy	Purchase
1	buy	Purchase
1	buy	Purchase
1	buy	Purchase
1	buy	Sale
1	buy	Sale

Order ID: 1

Name: buy

Order Type: Sale

7 Conclusion

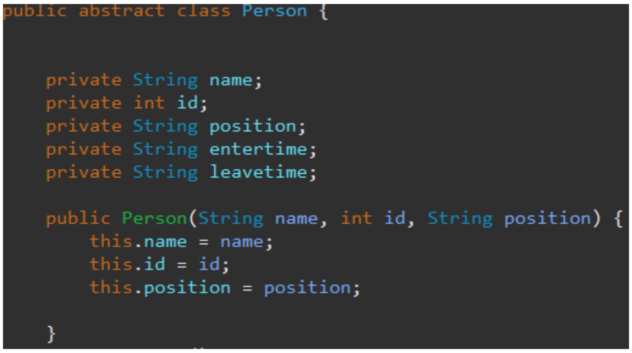
My project has been concluded and everything implemented as expected, queues used in storage unit as storage history. I have 43 classes total. In those 43, 19 of them are OOP codes ,4 is for queue implementation and rest of 20 are mostly the swing code.

8 DESIGN PATTERNS

In my project all 5 design patterns are in use which are Singleton - Factory - Abstract Factory - Prototype - Build patterns. Factory and Abstract Factory types sources are the classes can be seen in class lists on second page IReport Interface Class, Person Abstract Class, Order Abstract Class. Usages will be descibed one by one.

IReport interface is source for all system related reports generated Malfunction Report, Producement Report, Current System Report, Employee Report and lastly Vehicle Report these are used for reports created by system for corresponding names. Report interface consist of name, id, type and report details which can be filled in those classes according to system needs. This corresponds to the Abstract Factory Design Pattern.

Person abstract is source for all system related Users generated which are Worker, OtherEmployee, Manager, Employee and lastly Engineer. These people we address as Person all have shared information that can be generated as stock so that we used abstract class for this. Which they all extends.



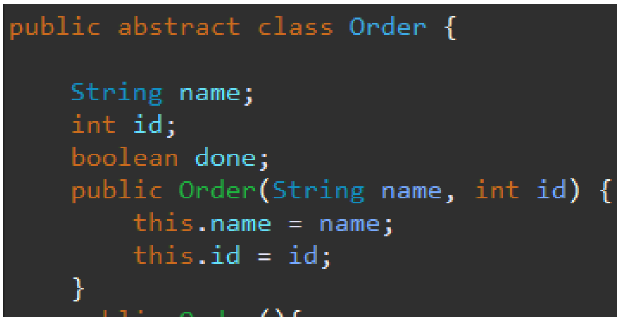
```
public abstract class Person {


    private String name;
    private int id;
    private String position;
    private String entertime;
    private String leavetime;

    public Person(String name, int id, String position) {
        this.name = name;
        this.id = id;
        this.position = position;

    }
```

Order abstract is source for all orders all across the system. This factory have two types of common orders which are sales orders and purchase orders,
Sale and Purchase is extended from this and used all accross platfrom.



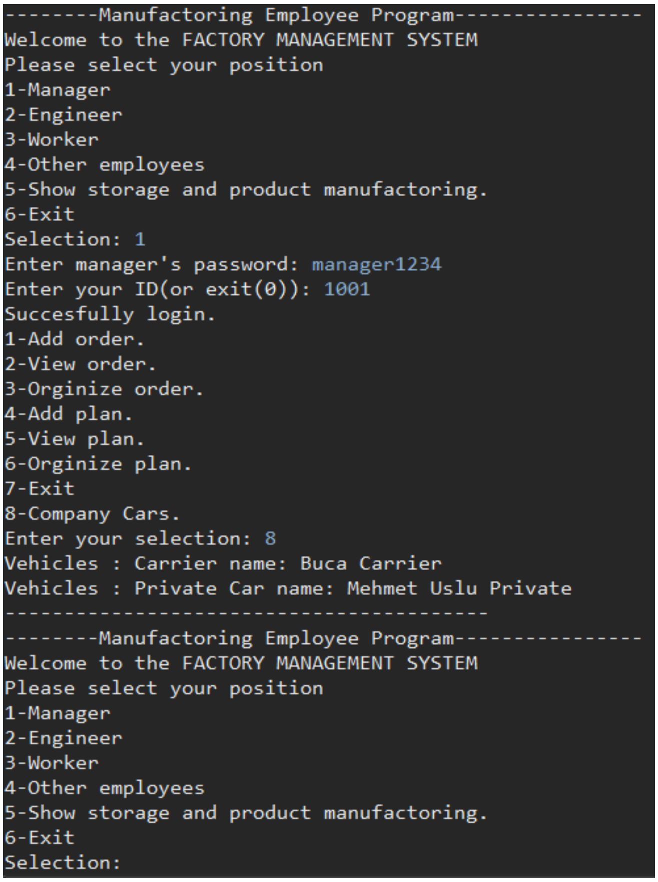
```
public abstract class Order {

    String name;
    int id;
    boolean done;
    public Order(String name, int id) {
        this.name = name;
        this.id = id;
    }
```

These two are corresponds to the Abstract Factory Design Pattern.

My Swing UI corresponds to the Builder Design Pattern because it completely connects all classes together.

Singleton is used for sending welcome messages on console UI and Prototype Design used for Vehicles that is designated to Company. Which can be buses, cars or lorries.



```
--------Manufactoring Employee Program---------------
Welcome to the FACTORY MANAGEMENT SYSTEM
Please select your position
1-Manager
2-Engineer
3-Worker
4-Other employees
5-Show storage and product manufactoring.
6-Exit
Selection: 1
Enter manager's password: manager1234
Enter your ID(or exit(0)): 1001
Succesfully login.
1-Add order.
2-View order.
3-Orginize order.
4-Add plan.
5-View plan.
6-Orginize plan.
7-Exit
8-Company Cars.
Enter your selection: 8
Vehicles : Carrier name: Buca Carrier
Vehicles : Private Car name: Mehmet Uslu Private
----------------------------------------
--------Manufactoring Employee Program---------------
Welcome to the FACTORY MANAGEMENT SYSTEM
Please select your position
1-Manager
2-Engineer
3-Worker
4-Other employees
5-Show storage and product manufactoring.
6-Exit
Selection:
```

Welcome to the FACTORY MANAGEMENT SYSTEM message is coming from FactoryObject Class.

carCache is for to manager to see all company vehicles that have been added for test purposes.

To get to this screen first you need to add manager on console ui to do that you must enter admin menu for that you need to input -1 and passwords is deuceng<3 From there you add a manager and exit to main menu and select manager manager id will be 1001 because you added just one and password is manager1234. On that menu you can see Company Cars that is example for Prototype Design Pattern.