For Programming Patterns

Ever needed a system to keep inventory of your restaurant? This is precisely what we do with our project: The Stocking System

Project Report

By Kishaan Thalayasingham and Brian Arnautov

Contents

[Project Description 2](#_Toc184722790)

[Scenario 2](#_Toc184722791)

[Program Features and Screenshots 2](#_Toc184722792)

[Design Paradigms (Functionalities) 2](#_Toc184722793)

[Design Patterns 2](#_Toc184722794)

[Images 3](#_Toc184722795)

[Expected Outputs 7](#_Toc184722796)

[Challenges 7](#_Toc184722797)

[Learning Outcomes 7](#_Toc184722798)

[GitHub 7](#_Toc184722799)

# Project Description

## Scenario

Our project for programming patterns was to create a stocking system that any hypothetical restaurant could use. It is meant to simulate the process of ordering inventory for your restaurant, monitoring the order/deliveries, and receiving a response from a Supply Company.

# Program Features and Screenshots

## Design Paradigms (Functionalities)

* Sign in with users
* Create orders
* Receive deliveries
* View order and delivery statuses
* Storage system
* Etc.

## Design Patterns

1. MVC (Model View Controller) – The Whole Program
2. Singleton – Used for database and for the RestaurantController
3. Abstract Factory – Used for creating Product classes

## Images

Proof the database works

A screen shot of a computer

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

A screen shot of a computer screen

Description automatically generated Invalid input

A screenshot of a computer

Description automatically generated Valid Input

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated Can choose the type of product

A screen shot of a computer program

Description automatically generated Registers the input, but doesn’t accept it.

A screenshot of a computer screen

Description automatically generated These methods work, it will display the things once they are loaded in java from the database.

A screen shot of a computer

Description automatically generated

This method also works, just the database doesn’t have any loaded values there.

A screen shot of a computer

Description automatically generated

Exits successfully.

## Expected Outputs

* Text in the console
* Database Tables
* User input
* Other stuff

# Challenges

* Time Management
* Implementing internationalization (or lack thereof)
* Making a functional database
* Connecting the class methods and the database methods (especially due to the HashMap)

# Learning Outcomes

* Time management
* How to create a decent database in java (SQLite)
* How to properly implement HashMaps
* How to use GitHub properly
* The Intelli J IDE
* Better cooperation with teammates

# GitHub

<https://github.com/BrianAr17/Patterns-Project---Restaurant-Stock-System.git>