2020

STOCKing

Hüseyin Erdoğan

Table of Contents

1.INTRODUCTION

1.1 What the Problem is

As an innovation to customers ' existing systems the priority was chosen security in the stock system. Bosses who have access to all kinds of functions using the Admin channel will have powers such as adding and deleting employees, editing employee shifts, stock viewing, weekly, monthly and annual revenue accounts. At the same time, employees will also enter the system according to the tables of incoming customers ' requests. What is wanted will fall out of stock. It is the output principle of the system to easily see if employees are using the products.

1.2 Goals for the Project

The main goal in this program is providing the controling of bulk purchase and security for owners of business. As we control the income and outcome regularly, at the same time ensure the security cheap and easy way. Also there is a order system that keeps the data of customers' procurement. Every table of café / restaurant has a number and their order data are kept by their numbers. Stock system works synchronized with procurement system. For example if a customer orders any of the products , stock system works and number of this product's stock is decreased. On the other hand, stock control system checks whether this product's stock empty or not. Another feature of this program that provide a special page to owner of business. This page contains the income and outcome by daily, weekly and montly. When a employee enters a wrong data to system, only this page has the permission about fix this problem. Therefore this program ensure the security.

Our main target is the desktop application and it is planned to open to access via android tablets and phones later in the year.

1.3 Stakeholders

Several different types of stakeholders can be noted when it comes to our software. Local cafés / restaurants and markets would be main scope of the project. Convenience for shop owners as well as employees is one of our priorities.

In the future, the chain plans to move to restaurants, general shops, online clothing and dining applications.

1.4 Motivation for the Project

The fact that one of our team members used to work in a cafe leads us on some issues. He stated that his boss's biggest problem when he was working was because of the employees ' use of the products in stock and that this issue was a general problem. We, as a team, are focused on this problem.

1.5 Process Flow Preview

For our process flow, we intend to remain fixated on the headlines we initially talked to cafe managers about. The modeling process in its own right we feel is not a start to finish process. There will be times that we may have to go back to certain portions within the modeling activity to ensure a sufficient model. We wish to make sure that if we miss anything, we do not figure that out in the construction stage.

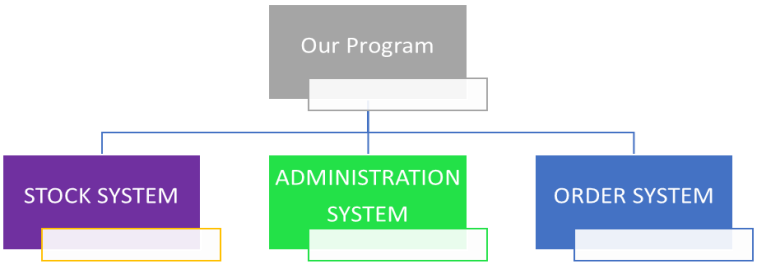
Thanks to our friend who worked in a cafe before and thanks to the environment we read and live in, we have acquaintances who own the cafe. They will help with the testing process, as we plan to talk to them one-on-one First and find solutions to their problems. We feel that allowing for the users to view how the software is made will ease the transition during the deployment stage.

2.ANALYSIS AND DESIGN

2.1 Plan for Requirements Engineering

Inception Task:

Our project appeals owners of local cafés and restaurants. According to information from the owners, our team has developed a program that has three main system; Stock System, Order System, Administration System. (Table 2.1)

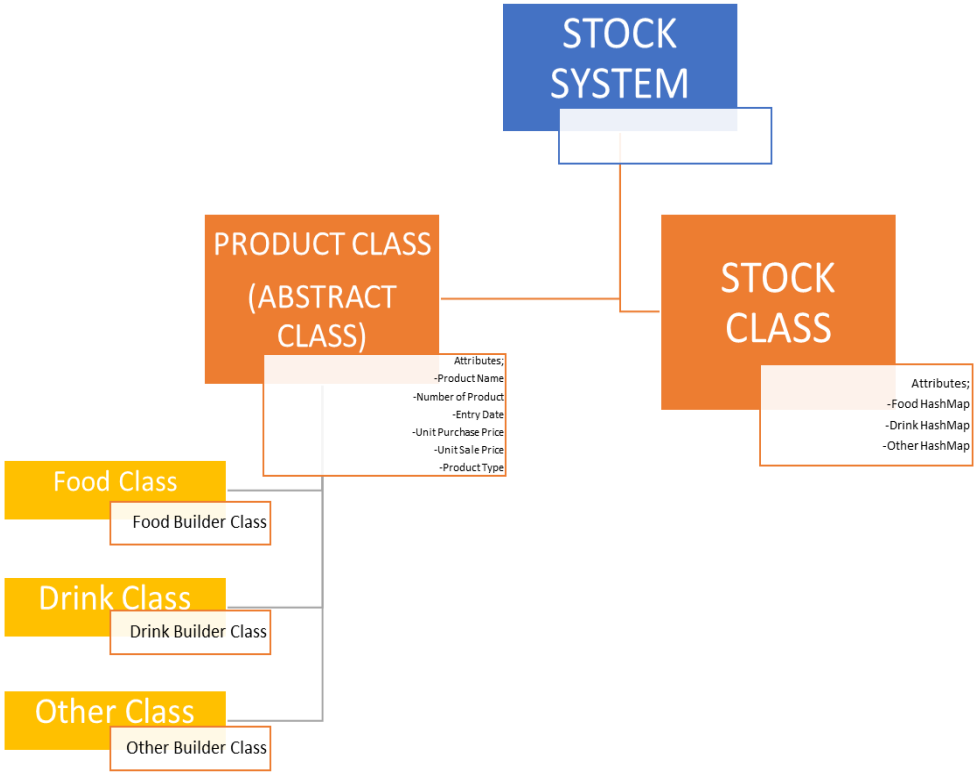


(Table 2.1)

Requirements Management:

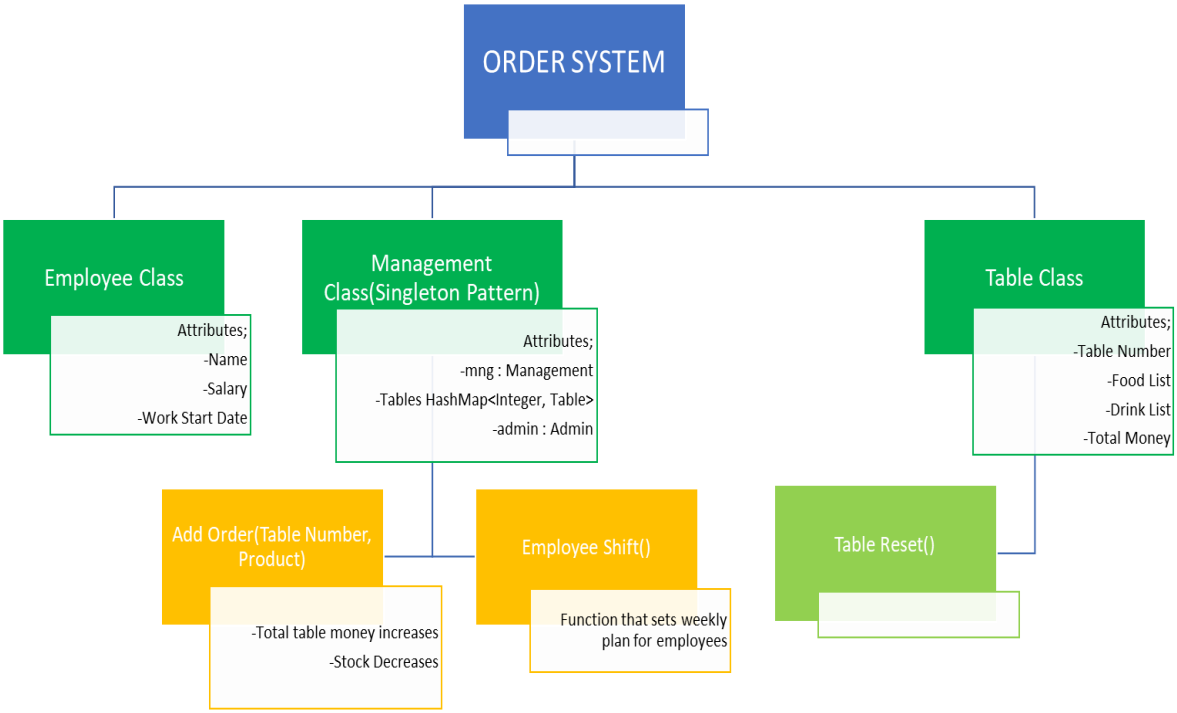
Any changes that may occur throughout the project stages should be handled with clarity and care. Any potential changes would be looked over, discussed and determined if the time allotted for the construction of the project can allow for such a change - that is if it is agreed upon by the stakeholders and software engineering team. Requirements Management will occur throughout the project process flow as changes or alterations can occur under any circumstances.

2.2 Functional Requirement



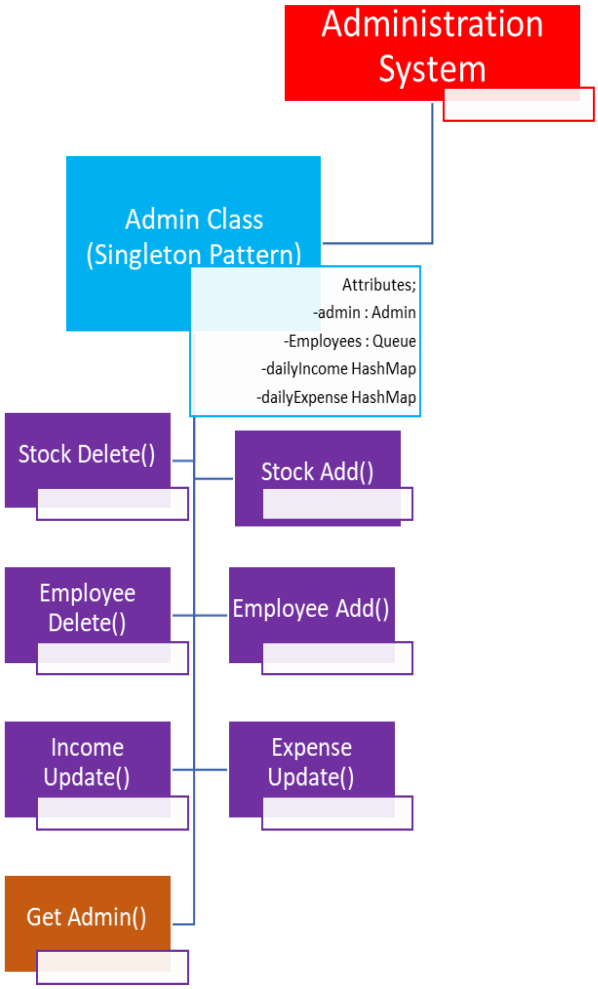
(Table 2.2)

First of all, our users request us to control the their storage and categorizing them. Therefore our team created the stock system. In this system, foods, drinks and others classes that inherited from an abstract class that is named products had been designed. In addition, these classes include builder classes. Builder classes give us a more effective way to define products and make our business easier. After that, there is a factory class called productFactory in our program. Creates the appropriate product using productType in the Product class. Lastly, stock class and also hashmaps in there had been created. Hashmaps had been prefered for reaching the products easily. (Table 2.2)



(Table 2.3)

Second requirements of our users were taking orders table by table works with stock system sync and employee's shift management. For this reason , table class, employee class and order class that would control first two classes had been defined. The Management class is the class where the general operations are performed and where the admin is present. Management class in this system are created with the singleton pattern structure. In this way, a single Management can occur and ensure that there is no process confusion. (Table 2.3)



(Table 2.4)

Last of them, our users demand us a administration system for security and easy access for all data. So that, product adding with product parameter to stock and removing from stock, displaying stock just for admins, employee hiring with employee parameter, employee firing, income and outcome data daily with graphs functions had been assigned.

The Admin class is created with singleton pattern, and in other classes, access is provided to the default admin with the getAdmin() function.

(Table 2.4

2.3 Non-Functional Requirements

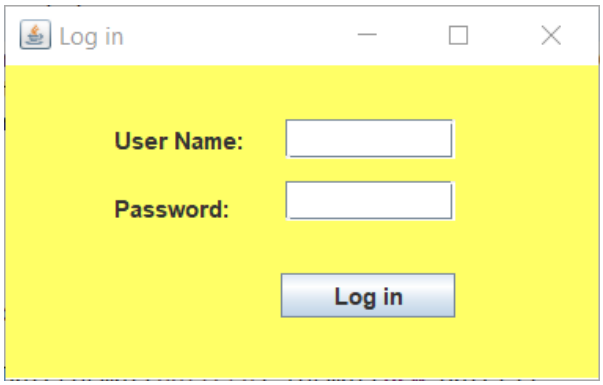
Security Requirements:

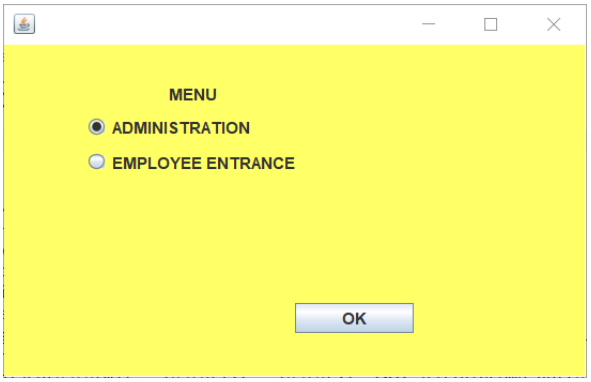
- Secure any transmissions of private information between the customer and the company
- Prevent third party users at administration level
- Prevent false information from being used as payment
- Prevent non-employee entry when logging in

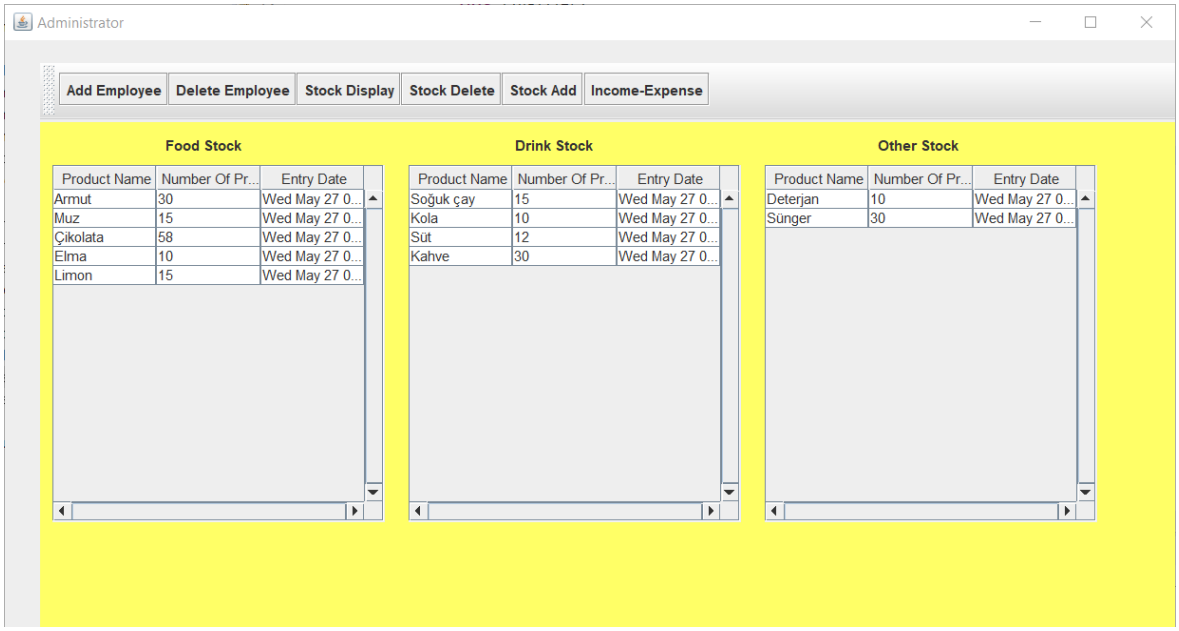
Quality Attributes:

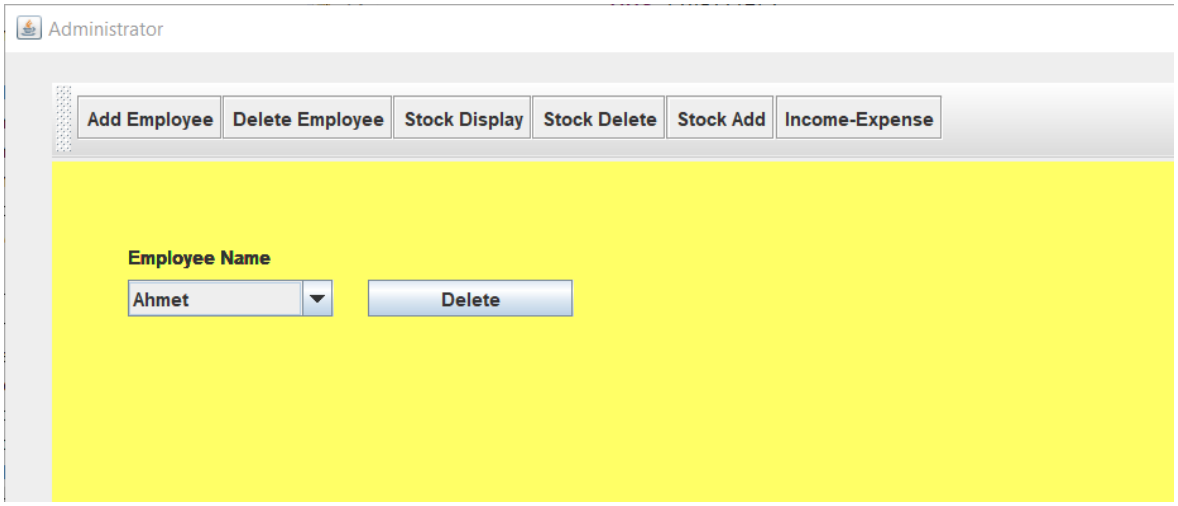
- Maintain a user friendly environment that is visually appealing
- Use warm colors and simple interface
- Easy to see and use navigation

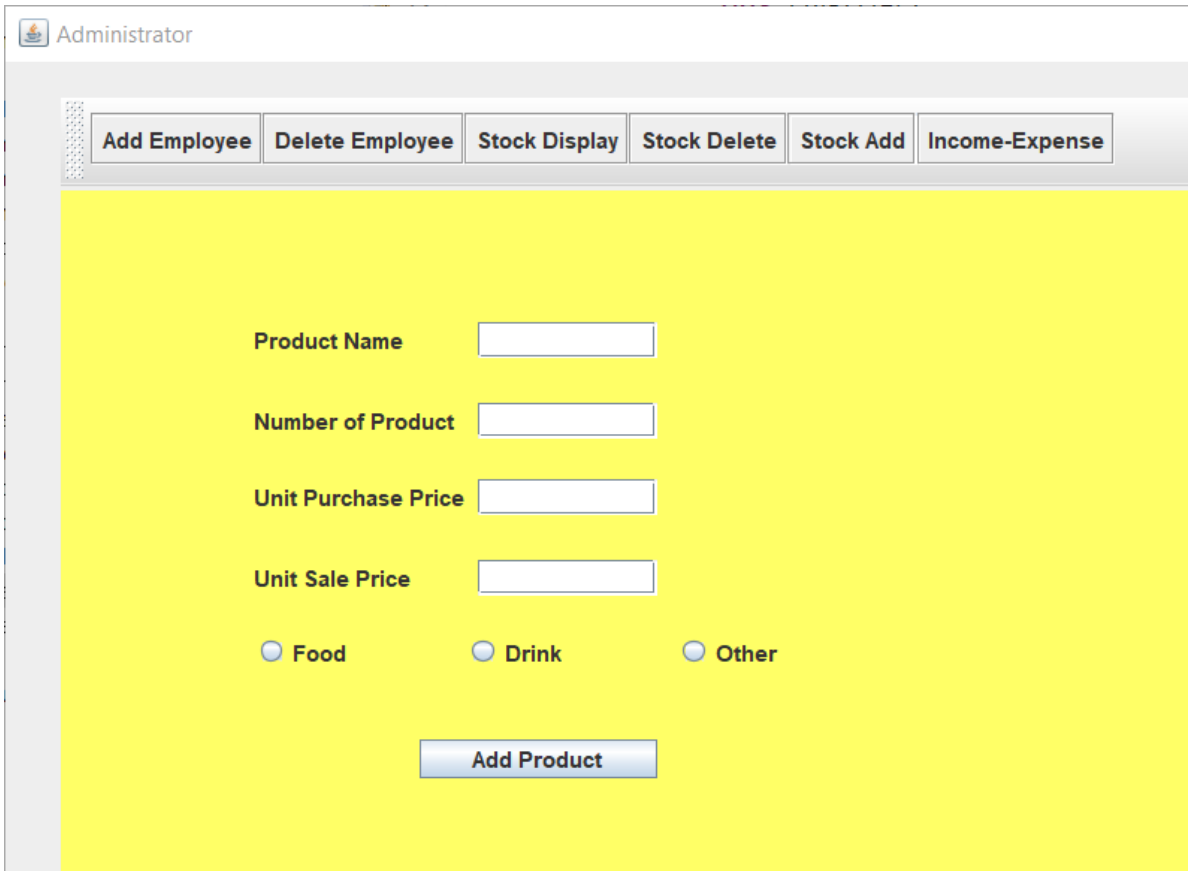
Screenshot Mockups:

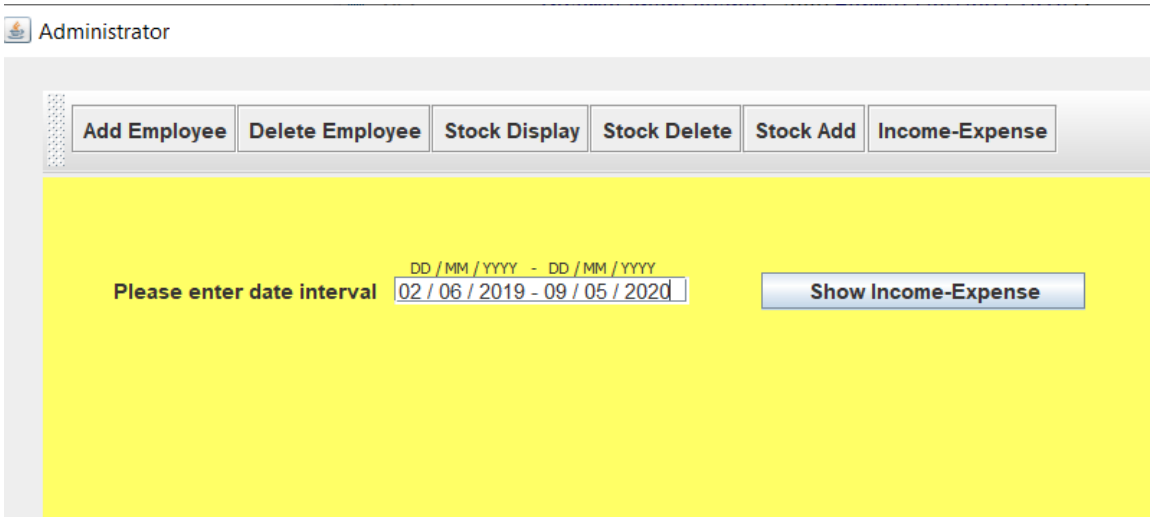


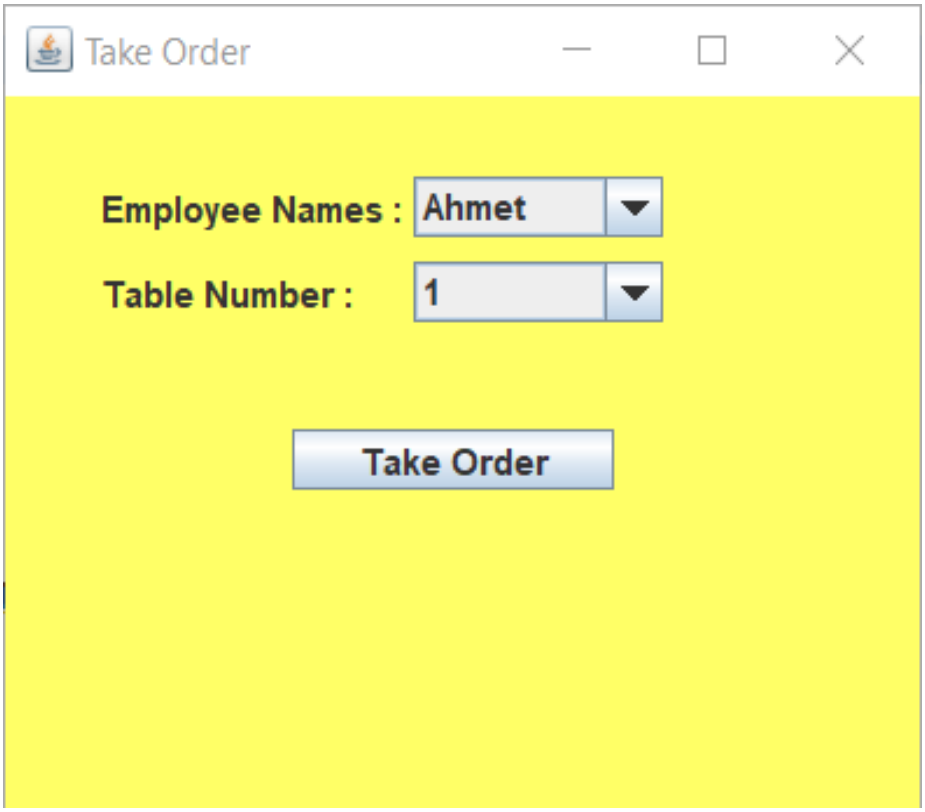


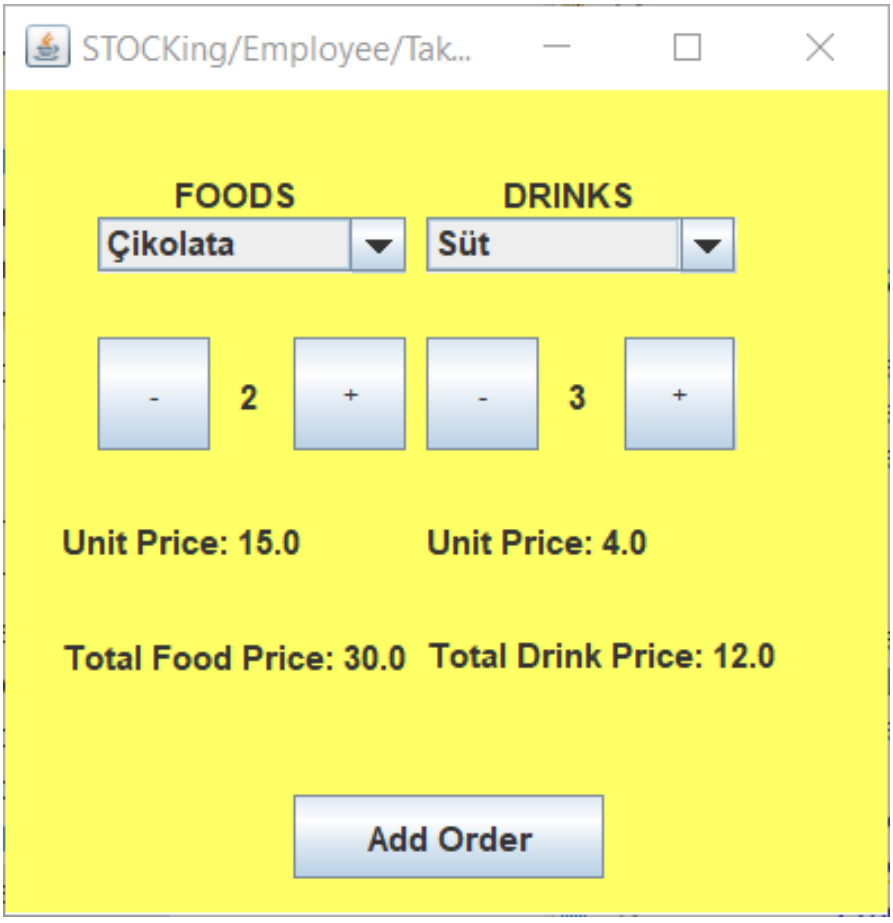

Administrator
Add Employee
Delete Employee
Stock Display
Stock Delete
Stock Add
Income-Expense
Food Stock
Product Name
Number Of Pr...
Entry Date
Armut
30
Wed May 27 0...
Muz
15
Wed May 27 0...
Çikolata
58
Wed May 27 0...
Elma
10
Wed May 27 0...
Limon
15
Wed May 27 0...
Drink Stock
Product Name
Number Of Pr...
Entry Date
Soğuk çay
15
Wed May 27 0...
Kola
10
Wed May 27 0...
Süt
12
Wed May 27 0...
Kahve
30
Wed May 27 0...
Other Stock
Product Name
Number Of Pr...
Entry Date
Deterjan
10
Wed May 27 0...
Sünger
30
Wed May 27 0...


Administrator
Add Employee
Delete Employee
Stock Display
Stock Delete
Stock Add
Income-Expense
Employee Name
Ahmet
Delete

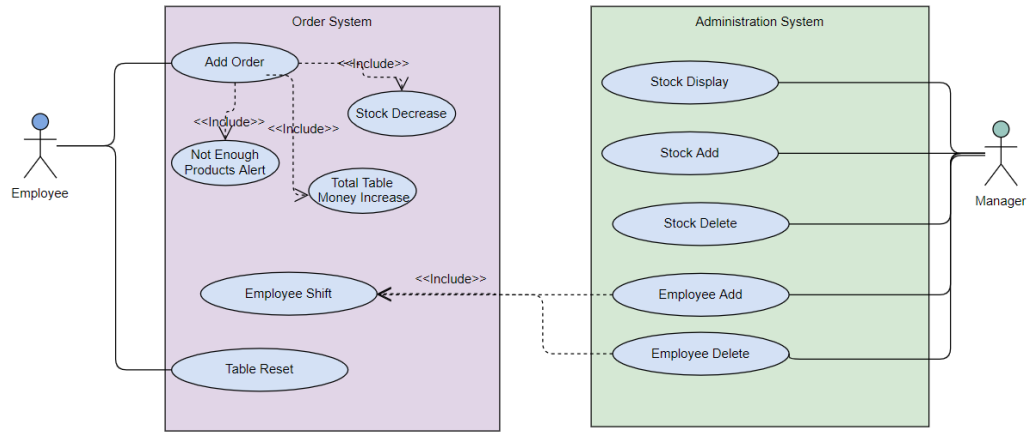

Administrator
Add Employee
Delete Employee
Stock Display
Stock Delete
Stock Add
Income-Expense
Product Name
Number of Product
Unit Purchase Price
Unit Sale Price
Food
Drink
Other
Add Product


Administrator
Add Employee
Delete Employee
Stock Display
Stock Delete
Stock Add
Income-Expense
Please enter date interval
DD / MM / YYYY - DD / MM / YYYY
02 / 06 / 2019 - 09 / 05 / 2020
Show Income-Expense


Take Order
Employee Names :
Ahmet
Table Number :
1
Take Order


STOCKing/Employee/Tak...
FOODS
DRINKS
Çikolata
Süt
-
2
+
-
3
+
Unit Price: 15.0
Unit Price: 4.0
Total Food Price: 30.0
Total Drink Price: 12.0
Add Order

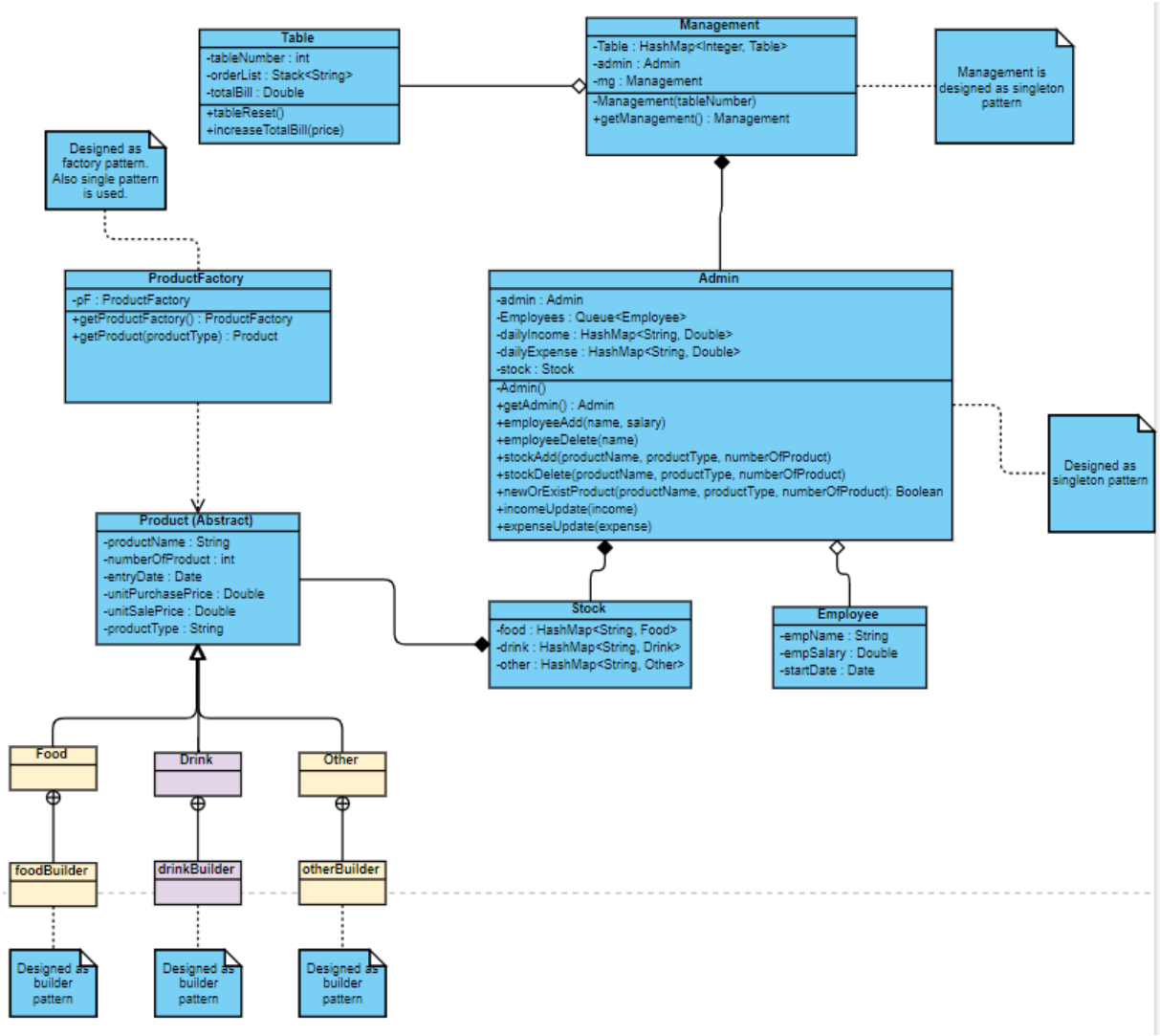
2.4 Use Cases



Use Case Diagram: In this project, there are 2 actors. One of them is employee and the other one is manager.

Employee can add order. Adding order causes decreasing of the stock, increasing the table bill and necessity of controlling the number of product. Because of these situations, 2 functions work: AddOrder (table number, product). Alert(product). Also an employee can reset tables.

Manager can display, add, delete the stock. Additionally, manager can delete and add employee. Therefore, employee shift function runs automatically.

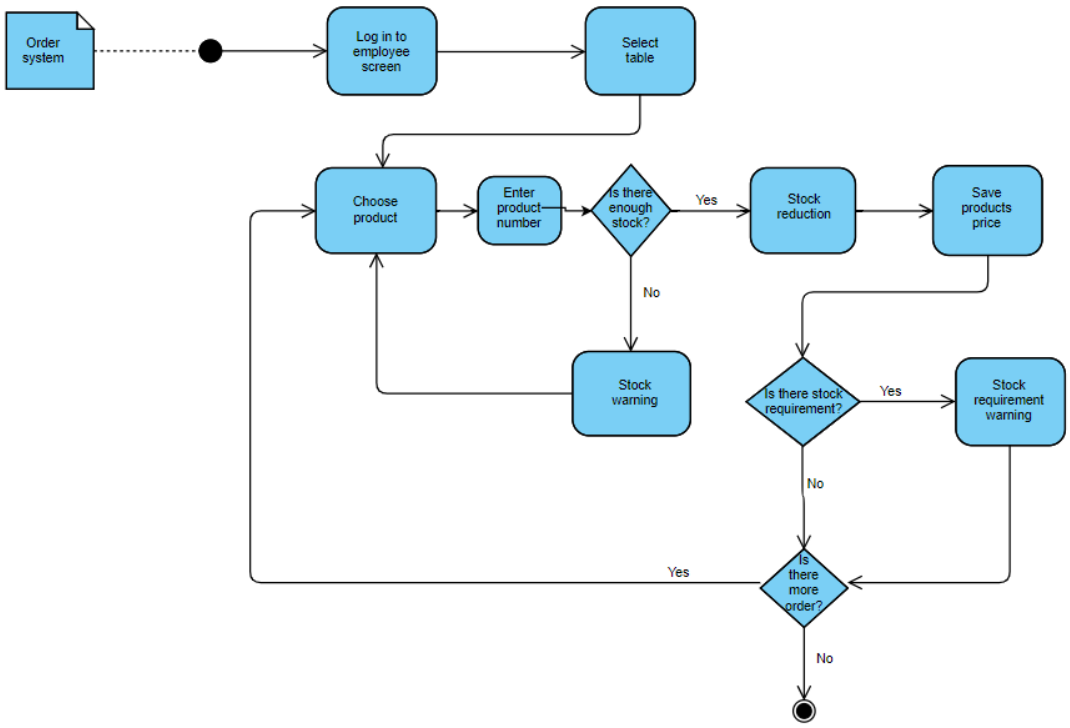


Class Diagram: To begin with, there is a product class in the program, which is abstract, where the products are generated in general. The classes connected to this Product class are Food, Drink and Other classes. In this way, the products are divided into 3 and there is one builder pattern for each. This builder structure gives us convenience in situations such as having missing data when creating products. Another class that works with the product class is the ProductFactory class, which is created with the factory pattern structure. he productFactory class is also designed and easy to call with singleton pattern. The purpose of this class is to create an automatically appropriate product according to its product type in cases such as adding a product to the stock.

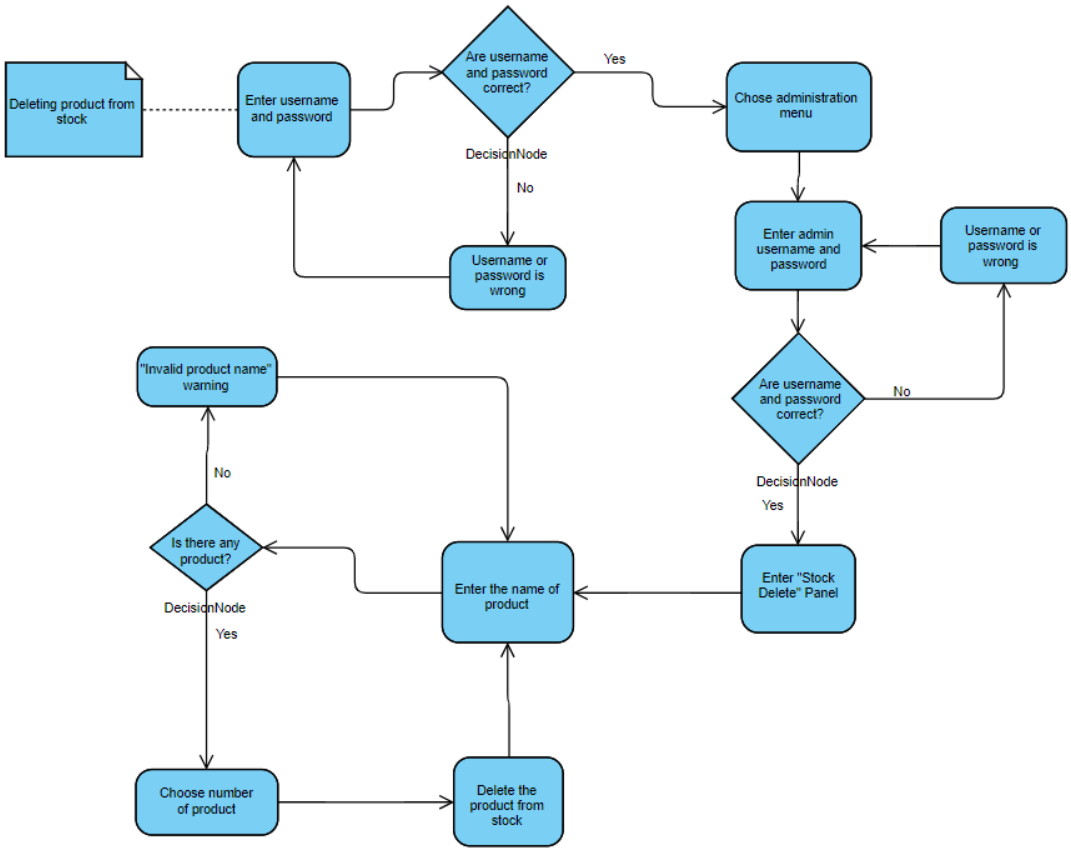
In general, the Admin class is the part where the data is kept, such as stock, employees. This class contains functions and information that may be useful to the administrator. As an example, we can give functions such as employee delete, employee add, stock display. Admin class has been created with singleton pattern design and it is planned to provide convenience for use in other classes. In addition, the stock and employee class are invoked and used in the Admin class.

The Management class is the main part where the Admin is created and the operations in the Admin class are controlled. The Management class is bound to the table class and the operations to be performed when the order is placed are performed in this class. In addition, the table class maintains the table number, order list, and total bill variables, and the order entered into the appropriate table is added under the control of the management. Finally, the Management class is designed with the singleton pattern design, providing ease for calling in the interface part of the program.

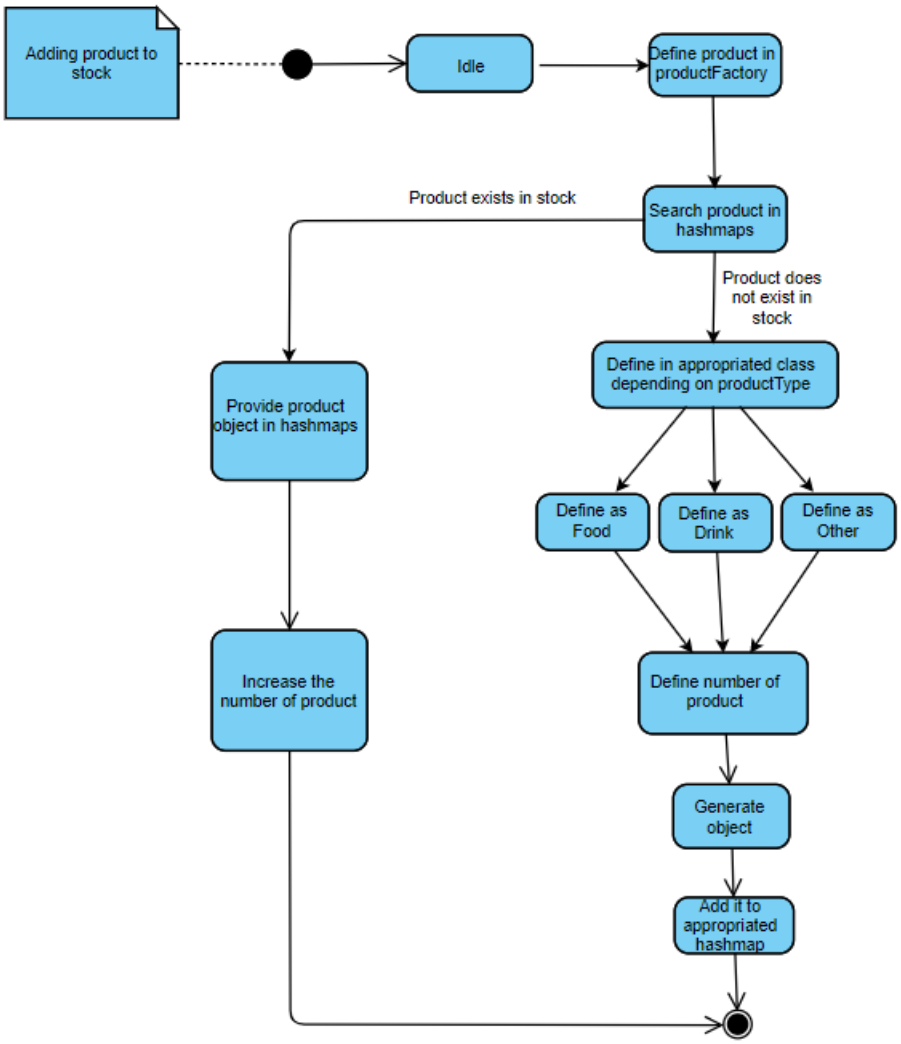
2.5 Models



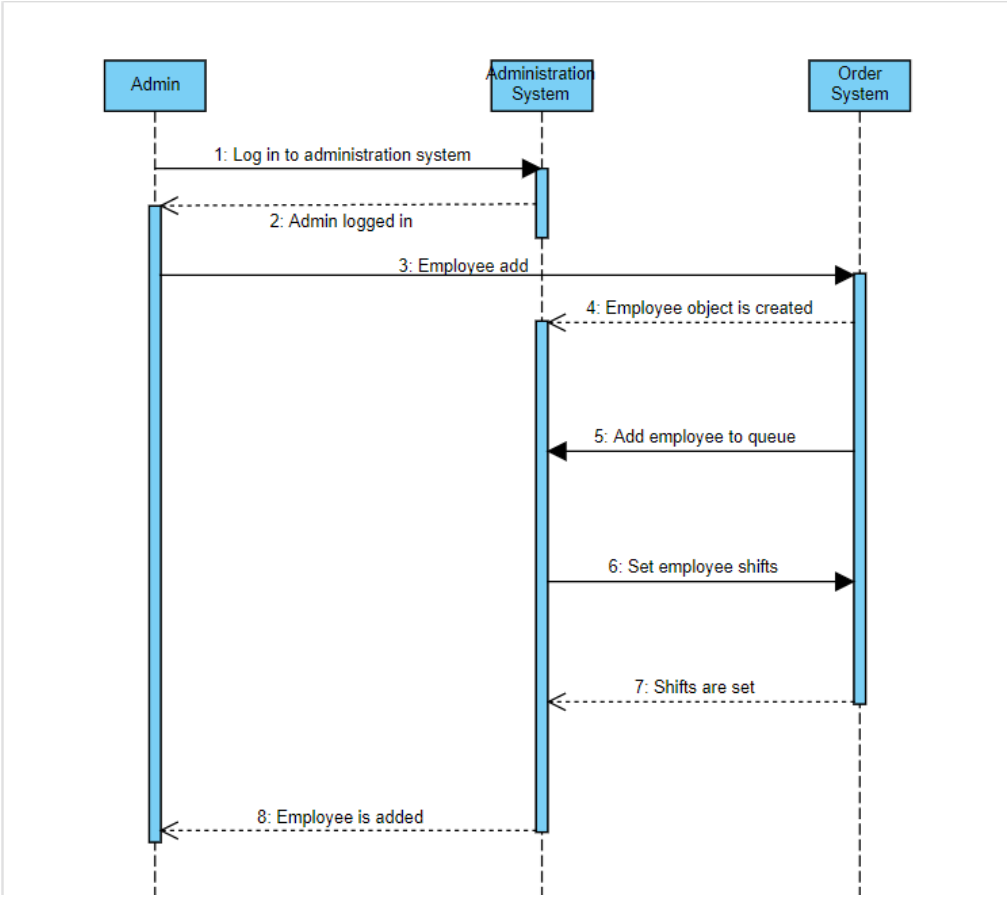
Activity Diagram-1 : At first, employees login to system and reach the employee screen. After that, the table that taken orders be chosen. Products and their numbers are entered to system. If there is not enough product in the stock, program gives an alert and returns to the product choosing screen. Otherwise, price of the current product is decrease from the stock and is added the table's bill. In other hand, program controls the stock for weather number of current product in stock is under the critical level or not. If there is this problem, program gives an alert for stock requirements. Lastly, program asks "Do you want to add another order?". If yes is chosen, returns choosing product screen. Other way, it exits.



Activity Diagram-2 : This diagram shows how product is deleted from stock. First, the user logs into the application and enters the username-password. in case of incorrect password, the program gives an error and asks for username-password again. After entering the correct password, the user selects which menu to go to in the panel that opens. User selects the "Administration Panel" for the Stock delete operation. Then, the admin password is queried again. After the correct password is entered, the "Stock Delete" panel is entered from the panel opened. Then, the program asks us for the product name and checks if the entered product is in stock. If it is in stock, the user selects the product quantity and ends the process of stock delete.



State Diagram: This state diagram shows us how to add products to stock. First, the product is defined according to the user's input information. This product identification process takes place in the class named" ProductFactory". Checks whether the product is in stock. If the product is already in stock, the number of items in stock is increased by the number of items entered. On the other hand, if it is not in stock, program identifies the appropriate product (Food, Drink Or and) according to the type of product and the product is created. Finally, the product is placed on the appropriate hashmap and the transaction is completed.



Sequence Diagram: Admin logins the administration system. Admin enters the information of the employees and object of employee is created in order system. This object is added to queue in administration system. Order system rearranges the employee shift automatically. In the end, admin receives the employee adding information from the administration system.

2.4 Design Patterns

Factory Pattern: The ProductFactory class was created primarily to use the factory pattern. The getProduct function in this class takes the information necessary to create the product and the type of the product that will be created and returns the appropriate product. Thanks to this class type suitable product is created smoothly and prevents complexity. The main part where this function is used is the "Stock Add" panel in the interface section of the program. The user marks the appropriate information and product type in the "Stock Add" panel, and the ProductFactory class ensures product creation.

Builder Pattern: In this program, builder design pattern was used in Food, Drink and Other classes. Builder classes are built as an inner class. These builder classes provide flexibility during product creation (such as missing some information). After the information requested to enter the product in the program. The product identification process has been completed with the build function.

Singleton Pattern: Singleton design pattern has been used in multiple locations in this program. The most important of these is the Management class. In the Management class, the Management object is created as private and is called from other classes only with the getManagement function. This way, other classes do not need to take Management as a parameter. Besides Management, it is also used in Admin and ProductFactory class.

3.PROJECT PLAN

3.1 Task Descriptions

Stakeholder Meetings

The owners of the cafe where we met have been interviewed. Ideas and suggestions about the system were received. Our team has considered these views. A plan has been drawn up.

Software Creation

The software that is to be used by the employees and bosses will be designed using Eclipse Java using the guide of the mockups, requirements, and models. The software will act as a simple and easy to understand user interface to use.

Testing

Status scenarios will be created and the program will be tested. Any bugs or errors that occur will be identified and resolved.

Finalization and Reports

All testing and function processes are finalized at this stage. Reports will be created to ensure all information and functionality is clear in order to make the user manual and to help ensure employees can use the software with ease.

3.2 Task Assignment

Task assignments were evenly distributed among the group assigned to the project. All three worked together on project planning, writing code and writing reports.

Stock, Food, Drink, Other, Product, Admin, Table, Employee, Order, Date classes are designed. Product class is an abstract class and Food, Drink,Other classes inherited from it. Five atributes are created in Product. These atributes are product name , number of product, entry date, unit purchase price , unite sales price. Stock object has been created in Admin class and there are several functions that can operate stock. Adding to stock , removing from stock, displaying stock are some of the operation functions. On the other hand Admin class

can control the Employee class. This control means employee hiring, firing, shifting. Lastly on Admin class, there are several arraylists that keep daily, weekly, montly incomes. Admin object is generated in Order class. That is only way to reach stock. When an order receives, addOrder function works and stock decreases over admin class. There is a hashmap that help us to reach tables in Order class. Also there is a hashmap that help us to reach tables with their numbers in Order class.

3.3 Deliverables and Milestones

We had four major Milestones in this project:

1. Completion of Requirements Gathering.
2. Completion of Design and code.
3. Completion of Testing.
4. Completion of Reporting.

These milestones were all completed on schedule and yielded a Deliverable at the end of each.

3.4 Project Schedule

The first month of the project start date (end of the February) was used mainly for requirements setting and research. Through the last two weeks of May, we were able to begin designing. Around the same time, testing began and continued throughout the remainder of the project. We then wrote out the final report. We finished the project in late May.

4.TESTING

4.1 Features to be Tested

We will start by using both static and dynamic testing strategies. The static strategies will include reviewing the basics of the application whereas the dynamic testing is based on actual code execution.

The features we tested were as follows:

- To ensure that the application itself ran (Dynamic)
- log-ins worked efficiently and consistently (Dynamic)
- operation of the password system
- accuracy of employee salaries and dates(static)
- accuracy of inventory tracking (Dynamic)

4.2 Test Cases

The following are examples of test cases we implemented:

- The number must be entered into the text input
- Amount values should be displayed with correct currency symbols
- To select from the selection list, click on it
- All fields on page should be aligned properly and must match their colors

4.3 Testing Schedule

The testing should begin right after the project itself begins. Keeping up on testing will ensure that any mistakes are caught early and corrected immediately.

5. CONCLUSION

5.1 The Problem and Solution

The Problem stemmed from the lack of managers to dominate stock numbers, and from employees using this shortfall. The customers needed such a system that they could manage the stock system themselves and their employees could make order entry and out.

As a solution, it was decided that a single system should have 3 different branches. The administrator will be able to control the inventory and workers from the admin console and receive material feedback. Employees will be able to drop data purchased from the system and Bill according to desks. This software, as well as the cafe system, will restrict employees ' access to stocks and ensure the security of products in stocks.

6.USER MANUAL

6.1 Software Description

The desktop application is opened with a password and the boss and employee parts are working separately. If the user is an administrator, he / she can access the admin page using his / her own password, make employee and stock updates, and reach his / her income and expenses according to the date.

Employees can log in to their own pages, enter the products that customers want on the tables and create receipts.

6.2 How to use the Software

The program is designed to appeal to users of all ages and audiences with a very clear interface. No one who is not on the staff should reach the program. For this reason, the program is opened by asking the password is logged. From the page resulting from the correct entry of the password, the administrator and the employee continue to select their own pages.

The administrator logs in to the admin page using his / her own password and reaches what he / she wants from the employee add and remove, inventory update control, revenue and expense control tabs.

Employees are authorized to enter their own pages and add products according to their table numbers and take orders. Each product received is deducted from the inventory according to the date of purchase and added to the income and expense list.

6.3 Troubleshooting Common Problems

Choosing

In the employee or product selection screens, if you think the Auto-Writer product has been selected and try to take action, the program will not run.

You can choose which product you want and continue your process.

Adding Products

As a result of overloading during transitions between panels, the program may not find the products you type in the search section.

As a solution, go back to the admin page will fix the program.

Input Error

The program will not work if there is a number instead of a number and a number instead of a number where the user needs to log in.

The places where the number should be written are very clear in the program.