**Store management system**

**Members:**

Eeswara Anvesh Chodisetty,

Arjun Mukesh, and

Kavin Sivakumar

**Project summary:**

To make a management system to handle tasks like inventory management, staff scheduling, payroll management, and finance management.

**Objectives:**

* To make an manage inventory
* To manage employees
* To manage finances

**Definition of the problem:**

The Store is a complex environment where a variety of challenges present themselves. These challenges include things such as keeping track of inventory and updating it in a concrete manner, or scheduling breaks and duration of work for each valuable employee to ensure smooth management. Other key things including payroll management ensure that the resources of the company get distributed correctly and each transaction is kept track of. Similarly finance management is important to keep the demand-supply chain consistent. All these challenges can be tackled using Object Oriented Programming paradigms implemented in Java. The goal is to create a suitable backend interface for optimal performance of the store.

**OOPs concepts used:**

* Inheritance
* Data abstraction
* Encapsulation
* Dynamic binding
* Delegation

**Elaboration:**

* Inheritance - Inheritance is a fundamental Object Oriented Programming concept. Getting the properties from one class object to another class is known as inheritance. This concept involves having a parent class and child class so getting the properties of parent class object to child class object known as inheritance. This is used in the classes //Insert classname where inheritance is used.
* Data abstraction - Abstraction refers to the act of representing essential features without including the background details or explanations. The user does not get to know the inner workings of the application when data abstraction is used. //Write the implementations i.e name of the classes and usage.
* Encapsulation - The wrapping of data and functions into a single unit is known as Encapsulation. The data is not accessible to the outside world. This is performed using access specifiers like public, private and protected. //Write an example of the access specifier used in our program.
* Dynamic binding - Binding refers to the linking of a procedure call to the code to be executed in response to the call. The code associated with a given procedure is not known until the time of the call at run-time. //Maybe give an example of overriding/overloading from our program.
* Delegation - Delegation means that you use an object of another class as an instance variable, and forward messages to the instance

**Classes used:**

* Employee
  + Cashier
  + Janitor
  + Manager
  + Aisle workers
* Staff shifts
* Schedule
* Inventory
* Product
  + Groceries
  + Produce
* Finance

**Inheritance (if used ) – class diagram**

****

****

**References:**

1. Timothy Budd, “Understanding Object oriented programming with Java”, Updated Edition,Pearson Education, 2000.

2. C ThomasWu,“An Introduction to Object oriented programming with Java”, 4th Edition,Tata McGraw-Hill Publishing company Ltd., 2006.

3. CayS Horstmann, GaryCornell, “CoreJava Volume–I Fundamentals”, 9th Edition, PrenticeHall, 2013.

4. www.javaguides.net