**MILKSHAKE FACTORY**

**Contents**

|  |
| --- |
| 1. Introduction 3 |
| 1. Objectives of the Project 3 |
| 1. About the Project 4 |
| 1. Design Patterns 5  a. Creational Design Patterns   b. Behavioural Design Patterns |
| 1. Unit Testing 7 |
| 1. Actualization of the Project 8 |
| 1. Problem Faced 13 |
| 1. Conclusion 13 |

**Introduction:**

The project milkshake factory is made for diary people who exports the milk products like the milk, curd, cheese, butter and many more milk products which are necessary in our daily life. Why did this project needed? it’s because of functions, marriages, parties and for every festival we need more dairy products than usual since we gather as a group of families to celebrate these things. And to supply these products to towns from villages where we have the plenty of dairy products which are required and needed by the people in daily life activities. The main motto of this project is to login and buy the necessary products.

**Objectives of the Project:**

The financial assistance is extended for processing of milk with the following objectives.

i) To enhance the keeping quality of milk

ii) To avoid the economic losses to farmers by procuring the milk in time from them

iii) To manufacture various milk products as per market demand

iv) To provide quality products at affordable prices to the consumers.

 Recent study shows that USA is the largest producer of milk in the world which is nearly 10% of the world production. Various milk-based products which can be manufactured commercially in a rural industry are Cheese, Curd, Ghee etc. These products can be manufactured by low-cost traditional methods and machineries. This kind of industries can be located in area where abundance quantity of milk is available. The cost of these products would be considerably low compared to those of big companies. A good number of employment generations is possible with low investment.  Milk and its products are the essential item of daily life in our country especially majority of Americans are vegetarian and thus milk and milk products are indispensable to Americans. The per capita consumption was

122 grams. Per day in 1979 against an availability of 281 grams. Today per capita consumption as well as production quantity has increased and there is a need to make available milk and milk products to the people at reasonable price which can be attained only by setting small scale model dairy units in different milk producing areas to cater to the local needs. There is a challenge of multi-national companies paying major role in this industry. So, it would be better for rural industries to produce milk products with a traditional tinge.  Rural industries can market their dairy products through Co- operative marketing channels. So that their product will no way get compared with that of large industry Cheese – There are different types of Cheese produced commercially, but the process given below is of Cheddar cheese made from buffalo milk. The raw buffalo milk should be preferably fresh and bacteriologic ally sound. The standardized milk is pasteurized at 71º C for 5 minutes and subsequently cooled

to10ºC. The pasteurized milk is inoculated with suitable culture of desired quantity and milk is then held at 8-10ºC for about 12 hours. The milk is then transferred to sterilized cheese vats for further processing where the temperature is raised to 34-35º C by circulating hot water in the jacket. A 40 per cent solution of Calcium chloride is added @ 15 ml. per hundred litres of milk, which is followed by the addition of starter culture 1.5-2 per cent of the milk allowed to ripen until its acidity comes to 0.19-0.2 percent. Henson’s powdered rennet is used @2.5-3 grams. For hundred litres of milk. The renewed milk is allowed to set till the curd attains consistency similar to that required for cheddar cheese making. It is then cut into cubes, which is left undisturbed for five minutes. The curd is then cooked gradually to raise its temperature to 39º C and kept at this temperature for 10 minutes with constant stirring. The temperature of the content of the vat is raised another 10 minutes, which is then lowered to 34-35º C by circulation of cold water. The cooked curd particles are gathered at the end of the vat and allowed to settle down at the bottom of the vat. The vat is then covered with lid and its content left undisturbed for 8-10 hours, until the acidity of whey increases to 0.4-0.45 per cent, while temperature is maintained 34-36º C. The whey is drained off and curd block is then stripped into long pieces and passed through milling machine to get small cubes of desired size. The sliced curd is uniformly spread in vat and washed with hot water for 4-5 minutes taking care that the curd cubes should not float in water. The hot water is drained; washed curd is filled in hoops of 35 x 28x 10 cm. Size and then pressed. The block of cheese is then obtained, smeared with salt marsh and left in the cold Storage (5-10º C and 90 per cent relative humidity) for 48 hours. The block is turned once and smeared with salt as before at the end of 24 hours. After salt simmering and initial drying for 48 hours the cheese is immersed in 18 per cent brine solution prepared by mixing pasteurized whey and water in the proportion of 2:1 and calculated amount of salt. It is allowed to continue for 12-15 hours in humidity-controlled Room at 15-16º C and 10 percent R.H. During this process the upper surface of floating block cheese is sprinkled with dry salt on alternate days. The cheese blocks are then removed and left to dry at the same temperature for 2-3 weeks. Subsequently these are washed with water at 50º C, dried, Para finned and kept in cold storage for another 4-5 weeks for further ripening. Total time for ripening is about 8-9 weeks.

**About the Project:**

While we are entering into MilkShake Factory project we will be left with login message and the login credentials are already pre-taken by the username as user and the password as pass. After entering your credentials and you have to click the enter then you will be logged in and you can have the access to the whole options.

If we are not entered the correct credentials as per the given username and the password it will leave us with the error message as Invalid Credentials. And also, the compilation will also stops simultaneously.

If we had entered the correct login credentials, then we will have the message as login success. Then we are having the whole bunch of activities to handle like to select the products, count the products from the cart, products from the cart, buying the products which we had placed in the cart and logout.

If we had selected one of the products from the list provided in the project. That will be added to the cart and the count in the cart will be increased and we can buy the product whenever we want. We will be having a question to continue further to buy more any products or not. If we press y then it will continue, but if we press n it will pushes us from the project as well as it will logout us from the project simultaneously.

If we want to the see the list of the products that are in the cart, we have to select the option named as products in cart within the project which can be accessed by the specific number which was given beside that name. As we see it shows that the products which were in the cart and the details of the product will be displayed.

If we want to see the count of the products which were present within the cart which they were added by the users.

If the user wants to buy the products which were placed in the cart, then we have to enter the number beside the buy products in cart and then the message will be appeared as getting ready for payment and the billing done.

**Design Patterns:**

In software engineering, a **design pattern** is a general repeatable solution to a commonly occurring problem in software design. 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.

**Uses of Design Patterns:**

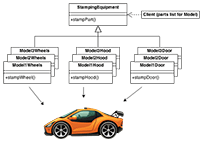
Design patterns can speed up the development process by providing tested, proven development paradigms. Effective software design requires considering issues that may not become visible until later in the implementation. Reusing design patterns helps to prevent subtle issues that can cause major problems and improves code readability for coders and architects familiar with the patterns.

Often, people only understand how to apply certain software design techniques to certain problems. These techniques are difficult to apply to a broader range of problems. Design patterns provide general solutions, documented in a format that doesn't require specifics tied to a particular problem.

In addition, patterns allow developers to communicate using well-known, well understood names for software interactions. Common design patterns can be improved over time, making them more robust than ad-hoc designs.

[**Creational design patterns**](https://sourcemaking.com/design_patterns/creational_patterns)**:**

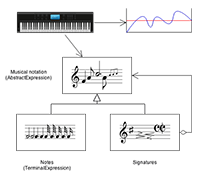
These design patterns are all about class instantiation. This pattern can be further divided into class-creation patterns and object-creational patterns. While class-creation patterns use inheritance effectively in the instantiation process, object-creation patterns use delegation effectively to get the job done.

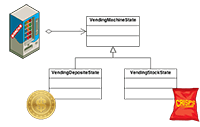
[](https://sourcemaking.com/design_patterns/abstract_factory)

* [**Abstract Factory**](https://sourcemaking.com/design_patterns/abstract_factory)  
  Creates an instance of several families of classes
* [**Builder**](https://sourcemaking.com/design_patterns/builder)  
  Separates object construction from its representation
* [**Factory Method**](https://sourcemaking.com/design_patterns/factory_method)  
  Creates an instance of several derived classes
* [**Object Pool**](https://sourcemaking.com/design_patterns/object_pool)  
  Avoid expensive acquisition and release of resources by recycling objects that are no longer in use
* [**Prototype**](https://sourcemaking.com/design_patterns/prototype)  
  A fully initialized instance to be copied or cloned
* [**Singleton**](https://sourcemaking.com/design_patterns/singleton)  
  A class of which only a single instance can exist**.**

[**Behavioural design patterns**](https://sourcemaking.com/design_patterns/behavioral_patterns)**:**

These design patterns are all about Class's objects communication. Behavioural patterns are those patterns that are most specifically concerned with communication between objects.

[](https://sourcemaking.com/design_patterns/interpreter)

* [**Chain of responsibility**](https://sourcemaking.com/design_patterns/chain_of_responsibility)  
  A way of passing a request between a chain of objects
* [**Command**](https://sourcemaking.com/design_patterns/command)  
  Encapsulate a command request as an object
* [**Interpreter**](https://sourcemaking.com/design_patterns/interpreter)  
  A way to include language elements in a program
* [**Iterator**](https://sourcemaking.com/design_patterns/iterator)  
  Sequentially access the elements of a collection
* [**Mediator**](https://sourcemaking.com/design_patterns/mediator)  
  Defines simplified communication between classes
* [**Memento**](https://sourcemaking.com/design_patterns/memento)  
  Capture and restore an object's internal state
* [**Null Object**](https://sourcemaking.com/design_patterns/null_object)  
  Designed to act as a default value of an object
* [**Observer**](https://sourcemaking.com/design_patterns/observer)  
  A way of notifying change to a number of classes
* **[](https://sourcemaking.com/design_patterns/state)**

[**State**](https://sourcemaking.com/design_patterns/state)  
Alter an object's behaviour when its state changes

* [**Strategy**](https://sourcemaking.com/design_patterns/strategy)  
  Encapsulates an algorithm inside a class
* [**Template method**](https://sourcemaking.com/design_patterns/template_method)  
  Defer the exact steps of an algorithm to a subclass
* [**Visitor**](https://sourcemaking.com/design_patterns/visitor)  
  Defines a new operation to a class without change

Here, from the above design patterns we have chosen factory method from the creational design patterns and from the behavioural design patterns we picked the command design pattern, iterator design pattern and observer design pattern to finish this project.

**Unit Testing:**

**UNIT TESTING** is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

In SDLC, STLC, V Model, Unit testing is first level of testing done before integration testing. Unit testing is a Whitebox testing technique that is usually performed by the developer. Though, in a practical world due to time crunch or reluctance of developers to tests, QA engineers also do unit testing.

## **Why Unit Testing?**

**Unit Testing** is important because software developers sometimes try saving time doing minimal unit testing and this is myth because inappropriate unit testing leads to high-cost Defect fixing during System Testing, Integration Testing and even Beta Testing after application is built. If proper unit testing is done in early development, then it saves time and money in the end.

Here, are the key reasons to perform unit testing in software engineering:

1. Unit tests help to fix bugs early in the development cycle and save costs.
2. It helps the developers to understand the testing code base and enables them to make changes quickly
3. Good unit tests serve as project documentation
4. Unit tests help with code re-use. Migrate both your code and your tests to your new project. Tweak the code until the tests run again.

## **How to do Unit Testing:**

In order **to do Unit Testing**, developers write a section of code to test a specific function in software application. Developers can also isolate this function to test more rigorously which reveals unnecessary dependencies between function being tested and other units so the dependencies can be eliminated. Developers generally use Unit Test framework to develop automated test cases for unit testing.

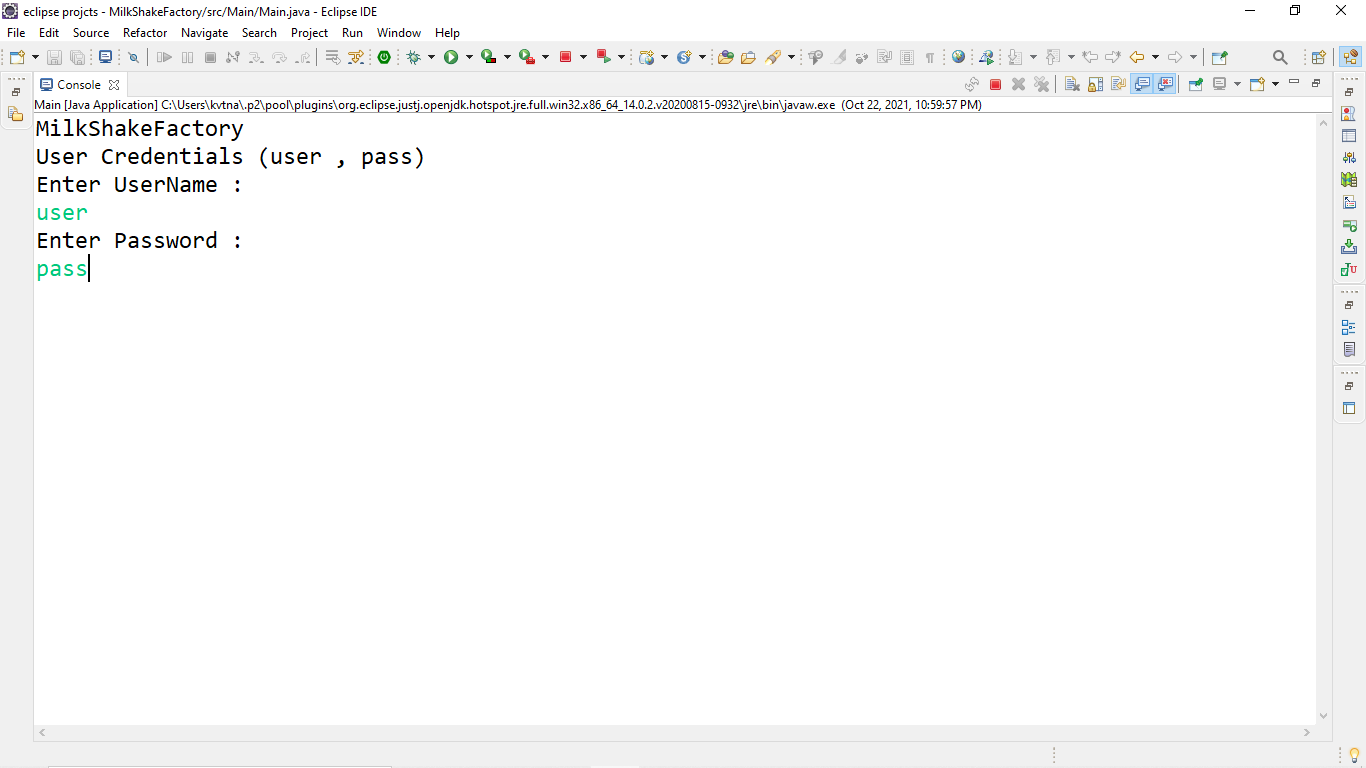
Unit Testing is of two types

* Manual
* Automated

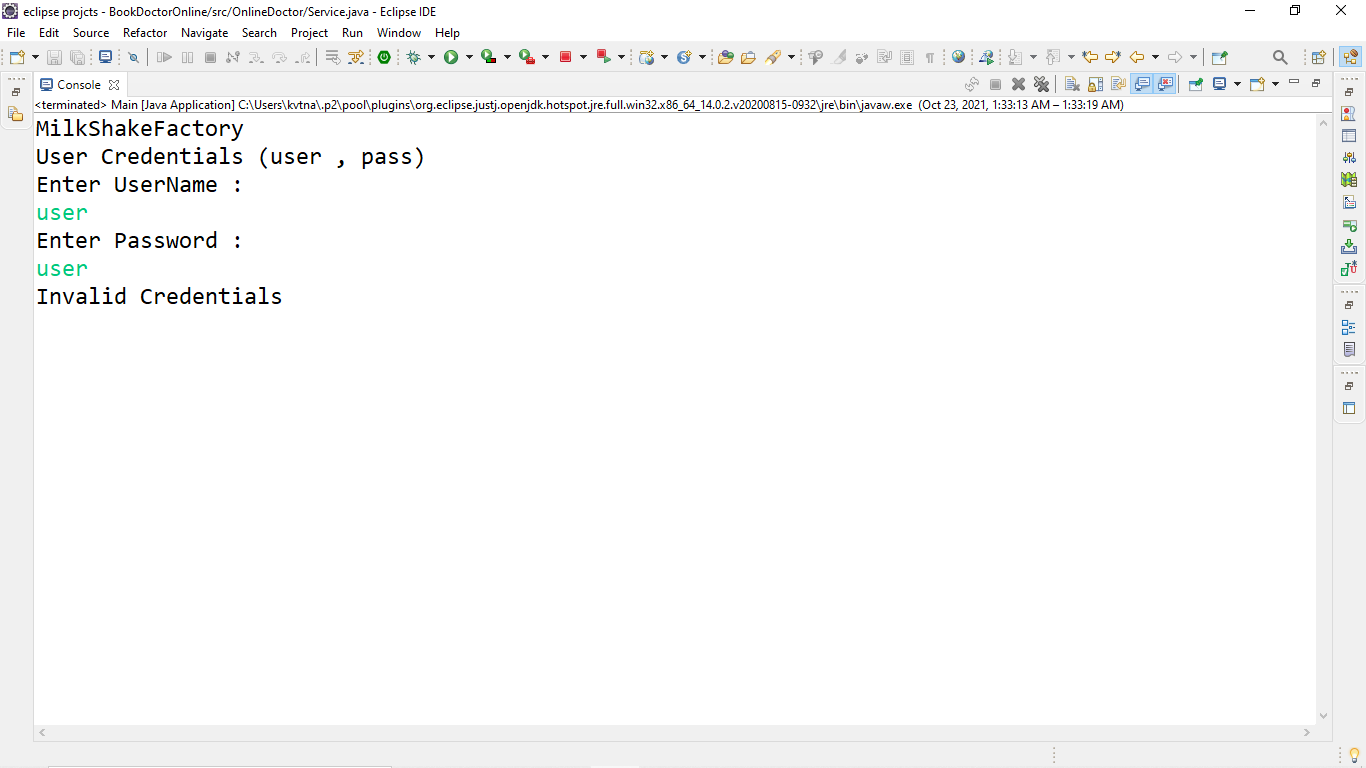
Unit testing is commonly automated but may still be performed manually. Software Engineering does not favour one over the other but automation is preferred. A manual approach to unit testing may employ a step-by-step instructional document.

**Actualization of the project:**

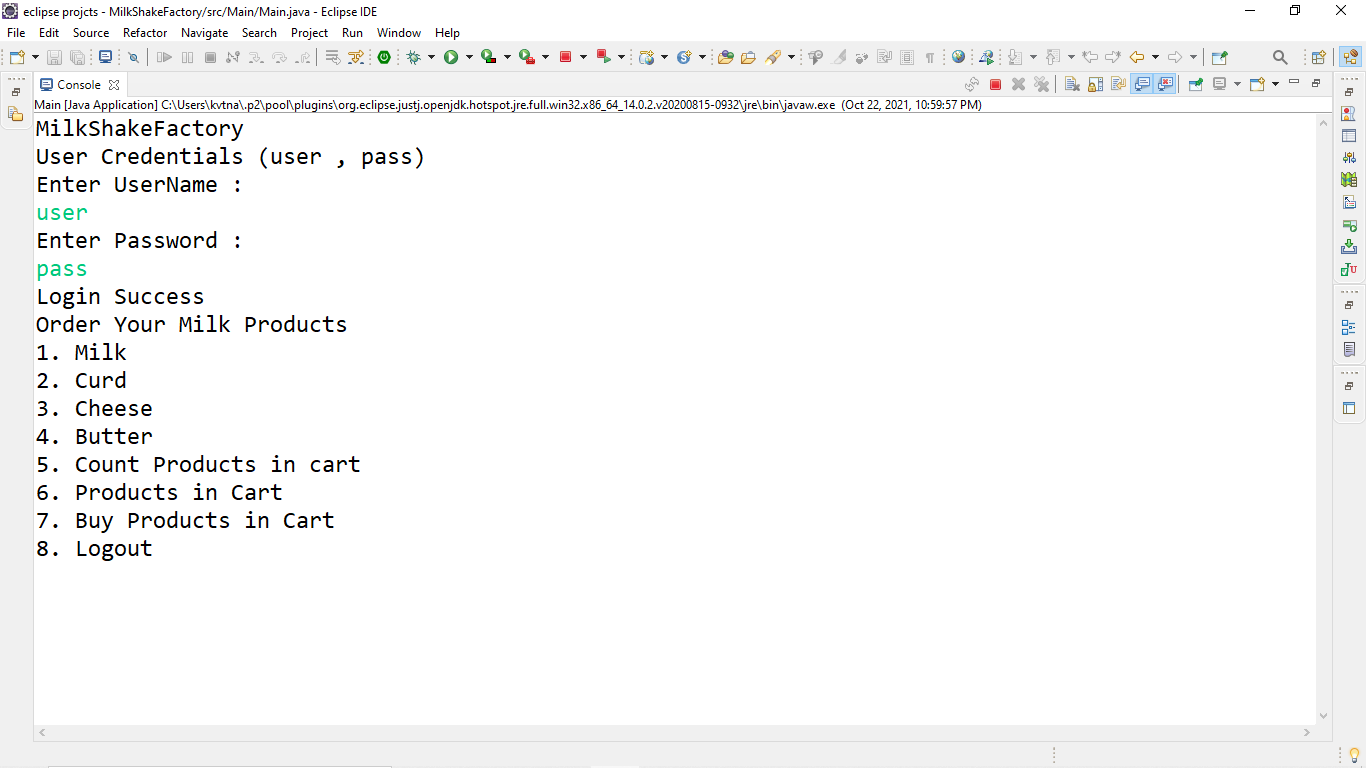
Running the MilkShake Factory project using console.



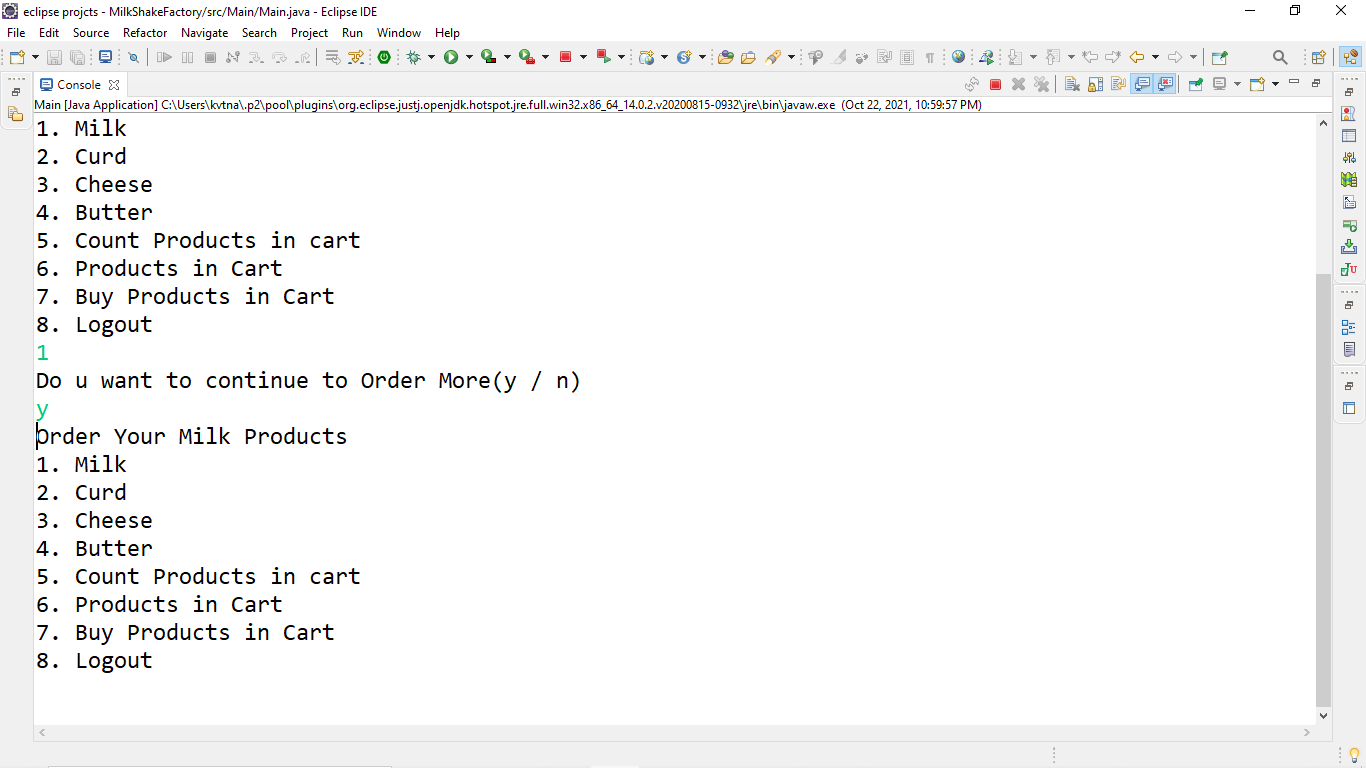
While we are entering in to MilkShake Factory project we will be left with login message and the login credentials are already pre-taken by the username as user and the password as pass. After entering your credentials and you have to click the enter then you will be logged in and you can have the access to the whole options.

****

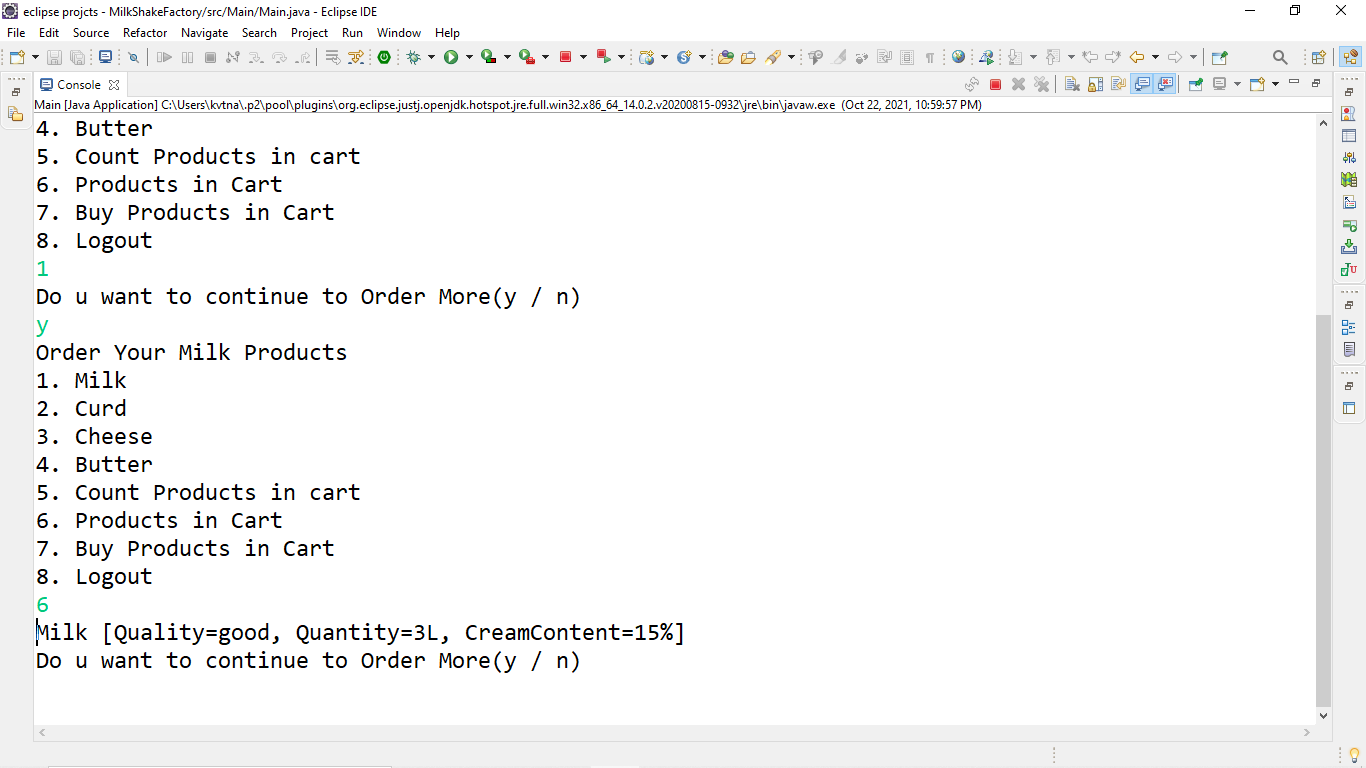
If we are not entered the correct credentials as per the given username and the password it will leave us with the error message as Invalid Credentials. And also the compilation will also stops simultaneously.



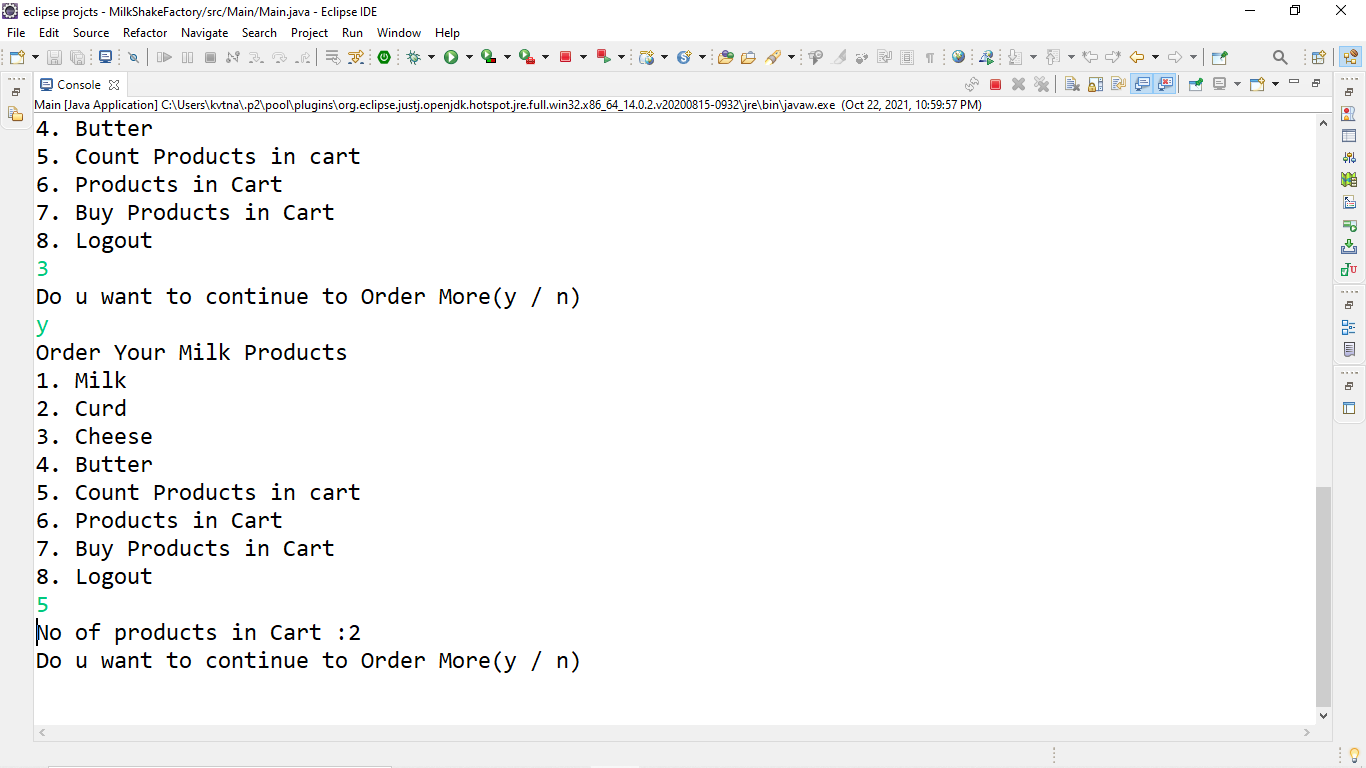
If we had entered the correct login credentials, then we will have the message as login success. Then we are having the whole bunch of activities to handle like to select the products, count the products from the cart, products from the cart, buying the products which we had placed in the cart and logout.



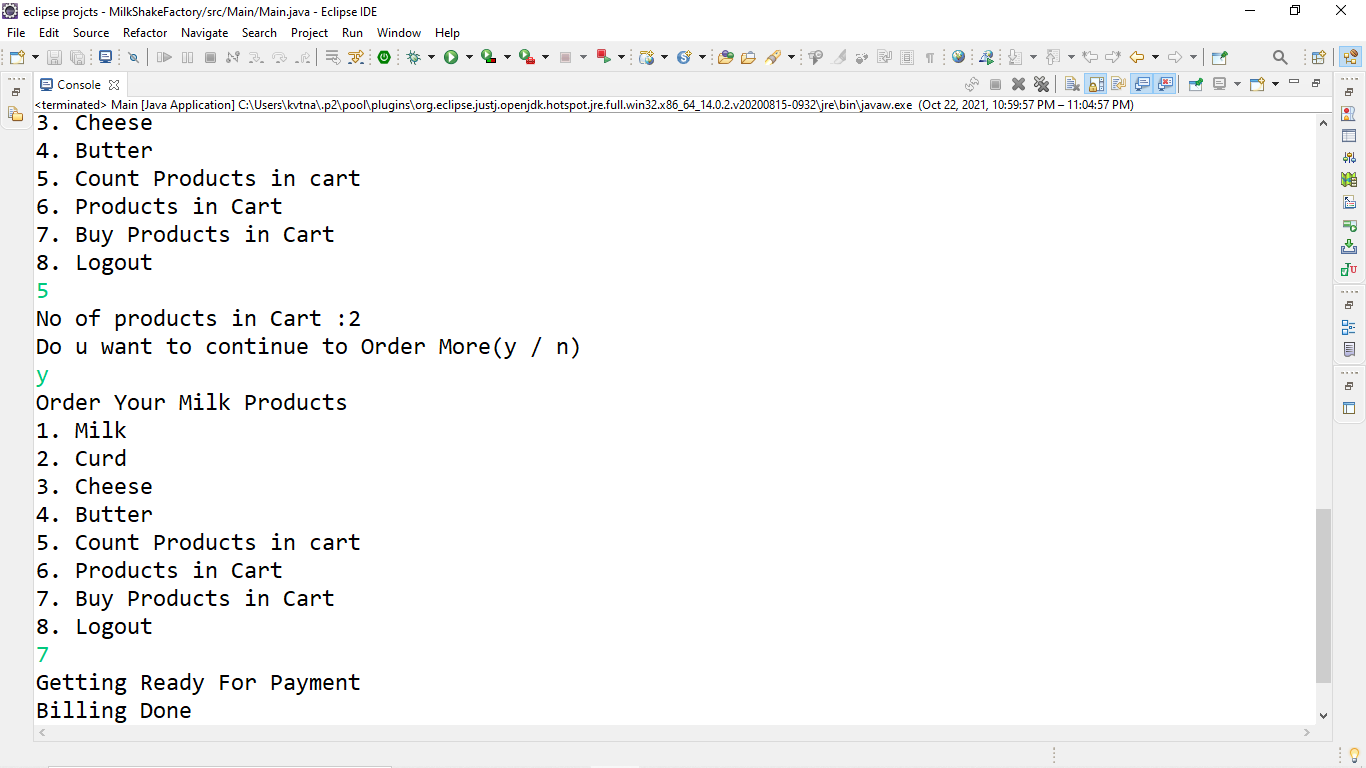
If we had selected one of the products from the list provided in the project. That will be added to the cart and the count in the cart will be increased and we can buy the product whenever we want. We will be having a question to continue further to buy more any products or not. If we press y then it will continue, but if we press n it will pushes us from the project as well as it will logout us from the project simultaneously.



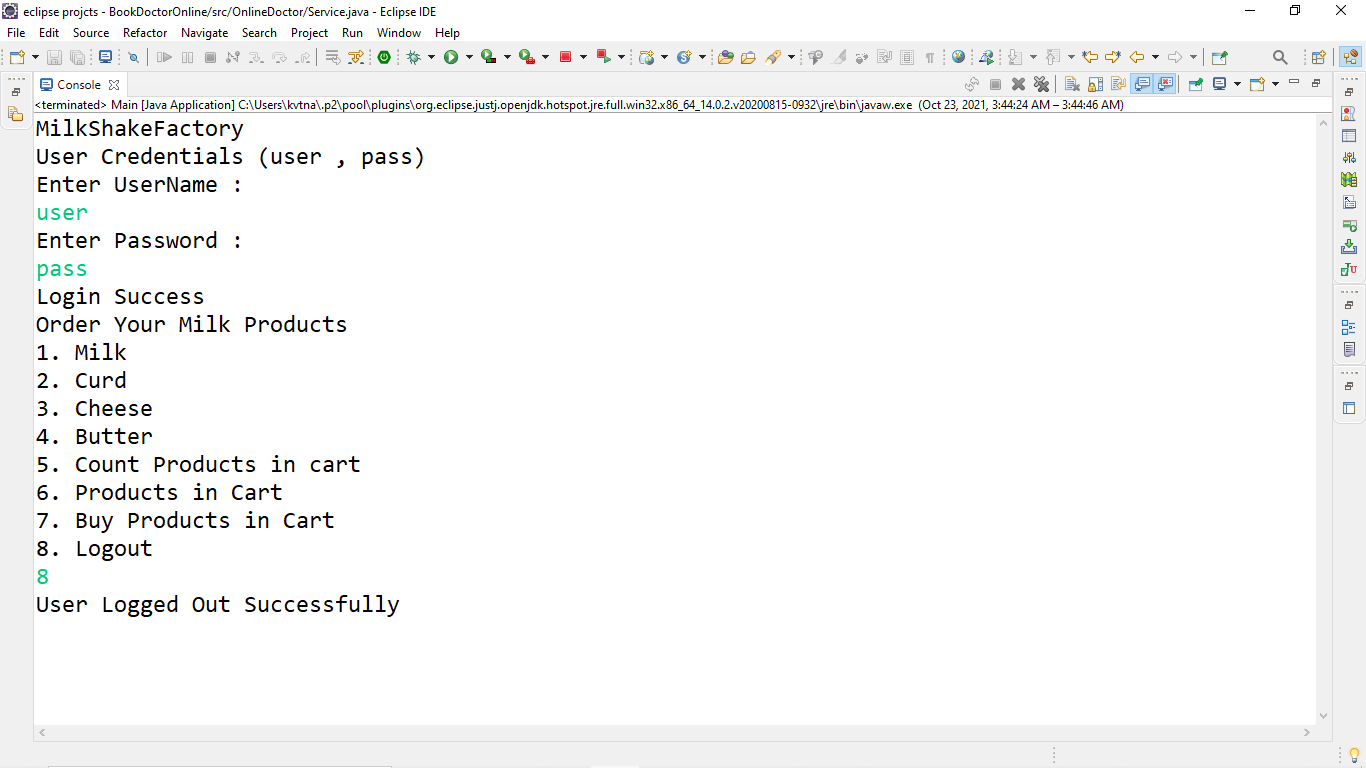
If we want to the see the list of the products that are in the cart, we have to select the option named as products in cart within the project which can be accessed by the specific number which was given beside that name. As we see it shows that the products which were in the cart and the details of the product will be displayed.



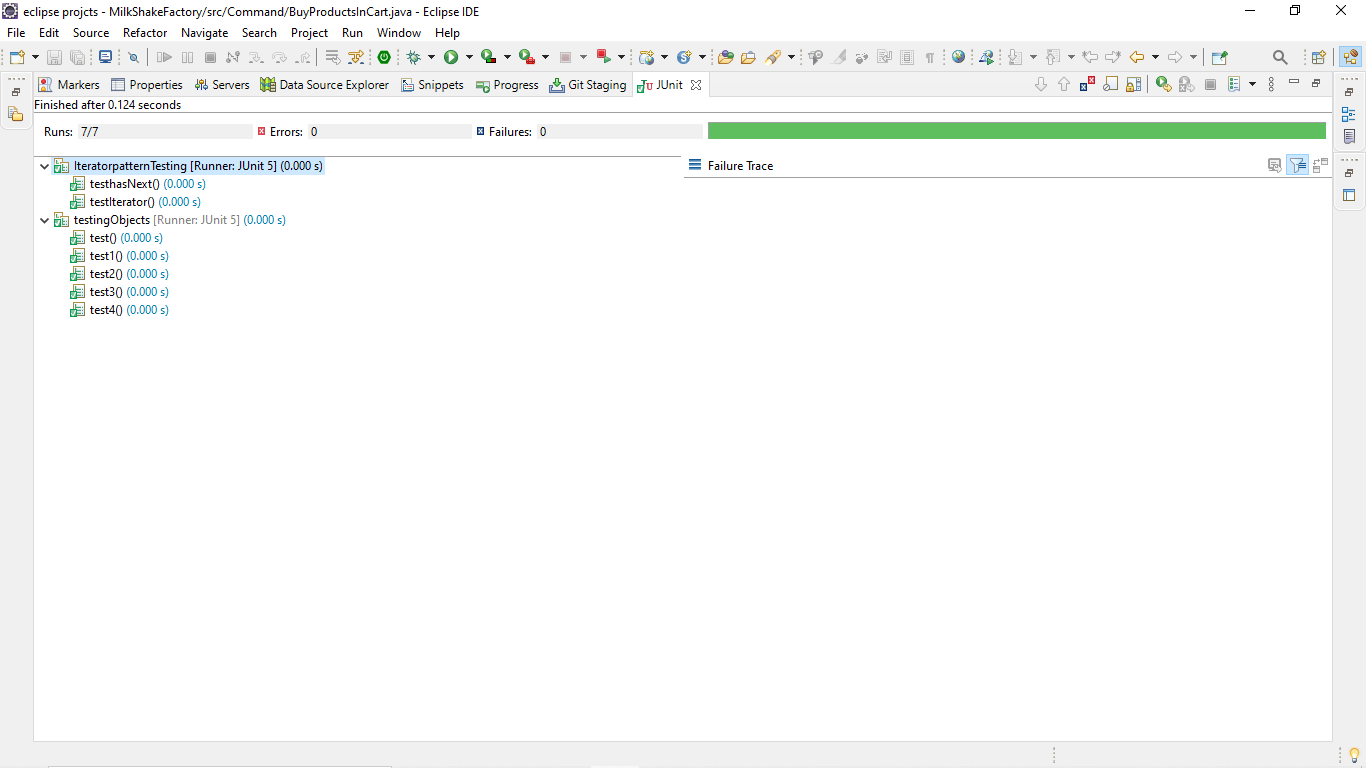
If we want to see the count of the products which were present within the cart which they were added by the users. As per the above screenshot it was showing the count of the products that were present in the cart.



If the user wants to buy the products which were placed in the cart, then we have to enter the number beside the buy products in cart and then the message will be appeared as getting ready for payment and the billing done.

****

If the user wants to logout from the project, then he we press the 8 that present beside the logout option given in the above list. Then the user will gets logout, and the project compilation will be stops.

****

Here, we have tested two design patterns and the result is shown in above figure.

**Problems Faced:**

* Adding the product to the cart.
* Counting the number of products.
* Combining the designing patterns efficiently.

**Conclusion:**

With the help of this project, we can conclude that the production for the dairy products is uplifted for higher production capacity.