|  |  |
| --- | --- |
| 1 | Abstract |
| 2 | Problem Definition |
| 3 | Business Requirement |
| **4** | Technical Requirement |
| **5** | Evolution Criteria |

**1. Abstract**

Modern coffee machines have become pretty complex. Depending on your choice of coffee, they decide which of the available coffee beans to use and how to grind them. They also use the right amount of water and heat it to the required temperature to brew a huge cup of filter coffee or a small and strong expresso.

We want to develop one Coffee Machine, where all coffee addicts can brew fresh cup of delicious coffee with their choices.

**2. Problem Definition**

We want to develop one Coffee Machine, where all coffee addicts can brew fresh cup of delicious coffee with their choice. The application should take in input from user for his choice of coffee & return a response displaying the cost & ingredients of the selected coffee.

User should be able to select a coffee of his choice.

**3. Business Requirement**

* Application will display all types of coffee available in the menu of choice to the user on console. The different types of coffee available are –
  1. Expresso
  2. CAPPUCCINO
  3. Mocha
  4. Filter Coffee
* Based upon the type of coffee selected by the user, we need to display a response on the console where the user can see the following details –
  1. The ingredients used in the coffee.
  2. The cost of the coffee.
* The various ingredients that can be part of the coffee are –

1. Milk
2. Sugar
3. Cream
4. Steam
5. Water
6. Coffee Beans
7. Ice
8. Chocolate Powder

**4. Technical Requirement**

* Application is console based application.
* Java 1.8 should be used.
* Must implement all OOPS principles(polymorphism, inheritance, abstraction, encapsulation, object , class) in entire application.
* Use Functional Interfaces, Lambda Expressions, Java Stream API and other Java 8 features.
* Use Spring JPA and Spring Data for data persistence.
* Use Spring 4.x, and you need to develop a project by using annotation only approach.
* Application has to be developed on micro service architecture,
* Use logging wherever it is required
* Use Maven.
* Create UML Diagram before implementation.
  + Class Diagram
  + Sequence diagram
* The whole application must be developed using test first approach. Test cases on each level are mandatory. TDD Approach.

**5. Evaluation Criteria**

* Modular Approach should be used in application development
  + Class level
  + Method Level
* Should make use of oops concept in appropriate situations.
* Should make use of oops principles in whole application.
* Should be using all clean code principles.
* Application code should be reusable
* Application code should be well testing using Junit and Mockito
* Appropriate names should be used for class, interface, method, variables, tables and columns
* Small method size, not more than 20 to 25 lines.
* User-friendly interface
* Accuracy of calculation
* Timeline delivery
* Completion status
* Sonar report will be generated from code.