

# FEATURES FUNTIONAL

Date	18 October 2022
Team ID	PNT2022TMID43271
Project Name	AI based discourse for Banking industry
Maximum Marks	8 Marks

## Introduction

Features are the “tools” you use within a system to complete a set of tasks or actions. Functionality is how those features actually work to provide you with a desired outcome .For example, a basic requirement for most boarding schools is the ability to cleave types. In this case, “custom leave types” is the feature, and the process/level is the functionality.

Some providers may only offer basic options while other vendors may offer advanced options like assignments, terms and conditions, workflows and rules. When you start to think about functionality, you’ll be able to gain a better understanding of what the system can actually do instead of what it says it can do. This will put you in a better position to decide which system will be able to provide you with the most value.

## How to measure functionality:

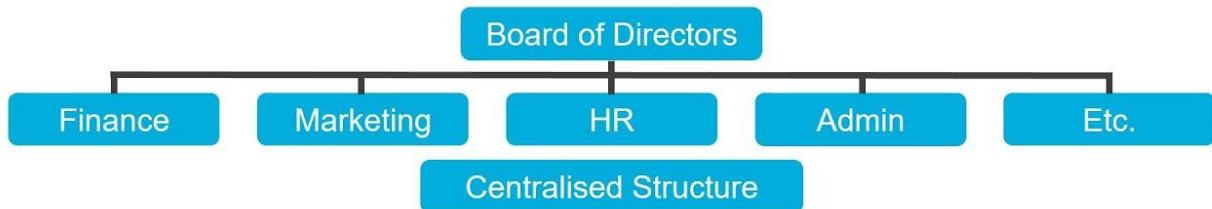
We use many different kinds of software products at boarding ware and have had to perform our own functionality assessments to make sure we choose the right one. Here’s our step by step process that you can also use when considering software for your boarding school:

1. Priority your requirements
2. Create a list of all the requirements and functions you need to solve your problems with boarding management. Then according to the value they will

bring to your boarding operations. This will allow you to focus more on the features and functions that will make the biggest difference and help you to which system will be able to provide you with the most value.

3. Get a Free Trial
4. Most software providers will offer a free trial period where you can properly test and evaluate the effectiveness of each feature. And if you have an list of functional requirements to test, you'll be able to make the most of your trial instead of just trying random features here and there. Boarding ware offers a structured 14-day free trial During your trial, go through your list of requirements and evaluate the following key areas for each feature:
  5. Test efficiency
  6. If the cloud doesn't simplify your job, it's not doing its job. Count the number of steps it takes to achieve your tasks and compare it with other systems or your current way of doing things. An extra click may seem like a minor issue but when you have to repeat actions for multiple students, the time to complete jobs can quickly pile up.
  7. Check quality/depth
  8. How well does each feature fulfill your requirements .Does it provide you with your desired outcome or is it missing important data and functions?
  9. Measure Intuitiveness / Usability
  - 10.If a product has a lot of functionality but is poorly designed and unusable, it's going to have very little value. The design is an important element that is often overlooked. Software products that have been well-thought out in terms of function and navigation will help to close the technology gap and make a system not just intuitive, but useable. A way to test this is to pay close attention to your first encounter with the system and see how long it takes you to complete a task or objective on your own

## Functional Structure



### Advantages

- **Experience**
- Specialist Staff
- Standardisation
- Greater Communication
- Economies of Scale
- Efficiency

### Disadvantages

- Lack of Variety & Job Enrichment
- Limits Flair & Ingenuity
- Greater Bureaucracy & Form Filling
- Compromise on Functional Requirements

### The problem with checking features:

Relying on a checklist of features is like looking at a car, seeing that it has 4 wheels, an engine, electric windows and a bunch of other cool features, and purchasing it without taking it for a proper test drive. It looks good on paper but when you actually drive the car you might notice that the electric windows don't work, the body kit is made of cheap plastic or even worse, there's a major engine fault.

## The Problem With Checklists



Wheels



Engine



Electric Windows



Brakes



Seatbelts



Power Steering



Checklists can give you a false impression of a product's functionality [caption] the same principle applies to software. While some software providers will deliver high quality features that offer a lot of value, others may cut corners and only provide minimal functionality just so they can check boxes. So it's important to remember that just because a product checks all the boxes on paper, it doesn't mean that it's going to work well in practice. To avoid any costly mistakes, make sure you take it for a test drive and check if those features are actually *Functional*.

```
// Java program to demonstrate
```

```
// anonymous method\
```

```
import java.util.Arrays;

import java.util.List;

public class GFG {

    public static void main(String[] args)

    {

        // Defining an anonymous method

        Runnable r = new Runnable() {

            public void run()

            {

                System.out.println(

                    "Running in Run able thread");

            }

        };

        r.run();

        System.out.println(

            "Running in main thread");

    }

}
```

**Output:**

Running in Run able thread

Running in main thread

Now, the above code has been converted into Lambda expressions rather than the anonymous method. Here we have evaluated a function that doesn't have any name and that function is a lambda expression. So, in this case, we can see that a function has been evaluated and assigned to a run able interface and here this function has been treated as the first-class citizen.

**Refactoring some functions from Java 7 to Java 8:**

We have worked many times with loops and so far up to Java 7 as follows:

**Characteristics of the Functional Programming Language**

Characteristics of the functional programming languages are like as shown below: Functional programming languages are designed to perform the functions of mathematical functions. These functions use conditional expressions and recursion to perform the computation .evaluation features .programming languages are designed on the concept of mathematical functions that use conditional expressions and recursion to perform computation.

- Functional programming languages are designed on the concept of mathematical functions that use conditional expressions and recursion to perform computation.
- Functional programming supports higher-order functions and lazy evaluation features.
- Functional programming languages don't support flow Controls like loop statements and conditional statements like If-Else and Switch Statements. They directly use the functions and functional calls.
- Like OOP, functional programming languages support popular concepts such as Abstraction, Encapsulation, Inheritance, and Polymorphism.

## **Conclusion**

As a downside, functional programming requires a large memory space. As it does not have state, you need to create new objects every time to perform actions.

Functional Programming is used in situations where we have to perform lots of different operations on the same set of data.

- Lisp is used for artificial intelligence applications like Machine learning, language processing, Modeling of speech and vision, etc.
- Embedded Lisp interpreters add programmability to some systems like functional.