# Test Strategy

## Types of Test

Functional testing is chosen for this project. For instance, boundary value testing is used as the method to do the test. The test goes through the development of this project and should update accordingly with any change on functionalities. Test must promise the final product is both robust and satisfied to specification. To ensure that, the strategy is performing TDD development. Generally, for each method in classes, multiple test methods need to be created according to the parameter types, boundary values and special considerations so that our test can cover all branches of the functionalities and disclose deficiencies as early as possible. For instance, test methods should at least cover the cases listed below.

|  |  |
| --- | --- |
| **Parameter Type** | **Test Value** |
| Numeric with a range | A valid value in range |
| Minimal value |
| Maximal value |
| Minimal value – 1 |
| Maximal value + 1 |
| Non-numeric, e.g. a character |
| A different type of number |
| Numeric without a range | A valid value |
| A different type of number |
| Non-numeric, e.g. a character |
| String | A valid string |
| Empty string |
| Null |
| A meaningless string values. E.g. invalid format. |
| Object | A valid object type |
| Null |
| An unexpected object type |
| A valid type of object without initialized value |
| List | A valid list |
| Null |
| An empty list |
| A list contains null element |
| A list contains unexpected class object |
| No parameter | Invoke once to check if result is valid |

## Testing tools and approach

JUnit is used as the testing tool. Junit is adopted to create test classes which contains test methods for each case. Test cases are developed first due to TDD development, then methods to implement functionalities are developed accordingly. After that, test cases are run automatically to check the validation of methods. Any fail case leads to modification of the corresponding functional method, after modification, all test cases need to be run again in case any new deficiency brought into project. Maven is chosen to help project’s build, version control, compile and run test cases. Maven is to auto run test cases.

## Defect tracking mechanism

To track defects, a spread sheet file is created to record all bugs occurred in the process of development. Here bug means both the errors or problems that lead to test case fail and any other problem result to unexpected behavior that against specification. The excel records the following information: Problem description, Test case affected, Reason of the problem, how to fix, fix time, status (open, in progress, fixed) and proof of fix.

# Setup Description

# Extension