# SYSTEM DESIGN

## INTRODUCTION

This document briefly explains system design of converting numbers based on roman numeral that will need to be implemented to cater for buying and selling over the galaxy.

## BACKGROUND

In order to fulfil the above requirement, following area will be added:

1. To create an application to calculate price of common metals and dirt from the galaxy.
2. To add new conversion algorithm.

## BUSSINESS RULES

Conversion or calculation will based on roman numerals. The numbers used for intergalactic transactions follows similar convention to the roman numerals. Roman numerals are based on seven symbols:

Symbol Value:

* I 1
* V 5
* X 10
* L 50
* C 100
* D 500
* M 1,000

Numbers are formed by combining symbols together and adding the values. For example, MMVI is 1000 + 1000 + 5 + 1 = 2006. Generally, symbols are placed in order of value, starting with the largest values. When smaller values precede larger values, the smaller values are subtracted from the larger values, and the result is added to the total. For example MCMXLIV = 1000 + (1000 −100) + (50 − 10) + (5 − 1) = 1944.

The symbols "I", "X", "C", and "M" can be repeated three times in succession, but no more. (They

may appear four times if the third and fourth are separated by a smaller value, such as XXXIX). "D", "L", and "V" can never be repeated.

"I" can be subtracted from "V" and "X" only. "X" can be subtracted from "L" and "C" only. "C" can

be subtracted from "D" and "M" only. "V", "L", and "D" can never be subtracted.

Only one small-value symbol may be subtracted from any large-value symbol.

A number written in Arabic numerals can be broken into digits. For example, 1903 is composed of

1, 9, 0, and 3. To write the Roman numeral, each of the non-zero digits should be treated separately.

In the above example, 1,000 = M, 900 = CM, and 3 = III. Therefore, 1903 = MCMIII.

## INITIAL UML DIAGRAM

UML stands for Unified Modeling Language. A class diagram models the static structure of a system. It shows relationships between classes, objects, attributes, and operations. There are three classes that we will create for this application.

1. MainClass, no attribute. Only contain one static method to pass in units name by StringArray. Purposed of this class is to run the whole application.
2. Conversion class, there eight attributes for units name and there one method/operation to process roman numbers conversion.
3. RomanNumbers, contain two attributes of class and one static method to calculate amount of units name given by user.

|  |
| --- |
|  |

## SERVER SET UP

1. Code uploaded into Github website.
2. Code implement using Java programming language. There are three files inside folder \GalaxyChallenge\src\com\mia.
3. MainClass.java is the main class to run and test the application.
4. Application develop by using Eclipse editor with jdk 1.8.