**Currency Converter**

**University of Maryland Global Campus**

**CMSC 495 7981 Current Trends and Projects**

Group 4: Justin Miller, Mike Yacht, Ayao Adanto, Brandon Tennyson

Project Analysis

Revision 7

March 30, 2020

**Revision Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Number** | **Description** | **Revisor** | **Date** |
| **1** | Initial Document | Ayao Adanto | 3/30/2020 |
| 2 | Team Review | Justin Miller, Michael Yacht | 3/31/2020 |
| 3 | Risk & mitigation | Michael Yacht | 3/31/2020 |
| 4 | Formatting and Grammar | Justin Miller | 3/31/2020 |
| 5 | Revise subsystem data diagram | Ayao Adanto | 3/31/2020 |
| 6 | Revised GUI description | Brandon Tennyson | 3/31/2020 |
| 7 | Revised risk and mitigation | Justin Miller | 4/24/2020 |

1. **Introduction**

The purpose of this document is to provide a detailed description of the Currency Converter program. It is intended to explain the purpose, the subsystems and the constraints of operation for potential developers in order to adequately design and implement Currency Converter.

**B. Analysis**

The purpose of Currency Converter is to convert one currency to another. It is a GUI-based program that shall allow the user to input a value of one currency and output its equivalent in another currency. The system shall require an internet connection. Currency rates will be updated frequently and stored in a database. Rates will be updated from **openexchangerates.org**, a currency rate web service.

1. **Outside Systems**

* User input to include text boxes and drop down menus.
* Database currency rates update from Exchange rate web API.

1. **Input data**

* Select target currency to convert from.
* Amount of starting currency.
* Select target currency to convert to.

1. **Output data**

System shall display in text areas:

* Target currency value
* Date and time of last update of database

1. **Data Processing**

During the startup, the Currency Converter program connects to the web API and retrieves the currencies’ rate to update the rates in the database and date/time of update. The system uses the rate stored in the database and user inputs to calculate the target currency value. When internet connection is not available, the system shall use available rates in the database to perform the conversion.

**Context Diagram**



**Figure1**

1. **Subsystem Components**

**5.1 GUI**

GUI (Graphic User Interface) allows users to interact with the system. System receives the currency value to convert through the GUI and displays the result on the GUI after computing the conversion. The GUI will have a textbox for the amount of the starting currency. It will feature drop down menus for the user to select the “from” and “to” currency. There will be a conversion button to activate the conversion. The GUI will display the output to the user.

**5.2 Currency Rate Web Service**

Currency Converter program retrieves the currency exchange rate from an API, **openexchangerates.org** to upload or update the database.

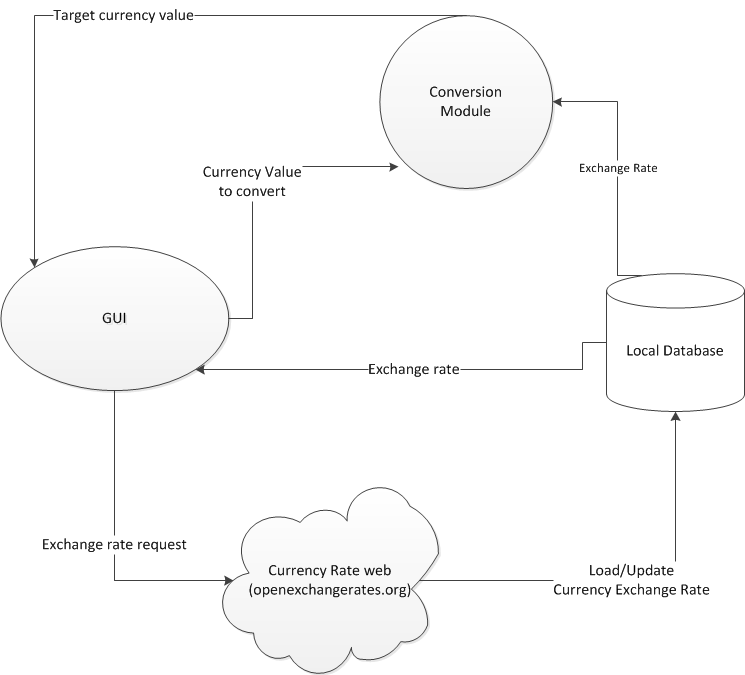
**5.3 Database**

The database is a flat file that stored the currency exchange rates. The database is uploaded and updated during the startup of the program before displaying the application GUI. Currency Converter program queries the database to retrieve the currency rate when the user submits the conversion request.

**5.4 Currency Conversion Module**

This subsystem calculates the equivalent currency value from data received from GUI and currency exchange rate retrieved from the database.

**5.5 Subsystem data flow**



**Figure2: Subsystem data diagram**

1. **Mapping of requirements to subsystems.**

|  |  |  |
| --- | --- | --- |
| # | Requirements | Subsystems |
| 1 | The program will obtain real time currency rates via the web | API, DB |
| 2 | Currency rates will be stored with a timestamp of their last update | API, DB |
| 3 | Base currency will be normalized on the USD | API, DB, Conversion Module, GUI |
| 4 | Shall require user to type value of currency in text box | GUI- Input Text Field |
| 5 | Shall require user to select the target currency type (to) | GUI - |
| 6 | Press a button “Convert” to calculate the target currency value | GUI - Button, Conversion Module |
| 7 | Display converted currency value (target currency) | GUI - Output Text Field |
| 8 | Accept decimal currency value (2 decimal point) | GUI, Conversion Module |
| 9 | The system shall support workstation-based GUI | GUI |
| 10 | The system shall have robust error checking and logging | Error checking (no included in subsystem) |
| 11 | Drop down menu for “from” currency | GUI – Drop down menu |
| 12 | Drop down menu for “to” currency | GUI – Drop down menu |
| 13 | Menus shall display 3 letter code and full name of currency. IE: US Dollars (USD) | GUI – Drop down menus |
| 14 | GUI shall display output in text area | GUI – Unmodifiable text field |
| 15 | Output shall be in format of input amount, from currency, =, output amount, to currency.  500.00 USD = 822.73 AUD | GUI - output Text field |

1. **Possible Enhancement**

* Store the last 5 years currency exchange rate.
* Provide historical exchange rates with graphs.

1. **Possible Risk and Risk Management**

* API Failure (could be for a variety of reasons)
  + Mitigation: Use latest previous conversion data
* No Internet Connection
  + Mitigation: Use latest previous conversion data
* Data type casting or overflow
  + - Try/Catch/Throw blocks to ensure data validation
* Data is aged and not updated
  + Date/Time is displayed on GUI for user to asses age of data.