|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | | | | | | | Currency Market Analyser Design Document | | | | | | |  | | |  |  | | | | Status: | final | | | | Version: | 1.0 | | | | Date: | **16 May 2019** | | | | Project ID |  | | | | | |
|  |  |  | |

Table of Contents

[1. Document Purpose 3](#_Toc8954848)

[2. Scope of solution 3](#_Toc8954849)

[3. Technologies Used 3](#_Toc8954850)

[4. Solution Architecture 4](#_Toc8954851)

[5. Details design of Logic for identifying profit 5](#_Toc8954852)

[6. Test Details & Screenshot 6](#_Toc8954853)

[7. ENHANCEMENT 8](#_Toc8954854)

[8. References 8](#_Toc8954855)

[9. Definitions 9](#_Toc8954856)

[10. Attachments 9](#_Toc8954857)

[11. DOCUMENT CONTROL SHEET 10](#_Toc8954858)

# Document Purpose

This document provides the Detail Design for new currency market analyser sample program.

# Scope of solution

User should be able to select a date and currency to view the maximum profit value and time gap in which profit achieved for list of crypto currencies available in stock marker.

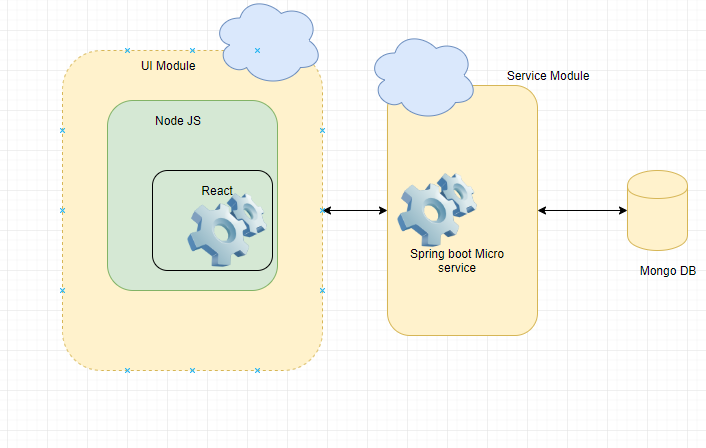
As an additional feature, users can select the time period (2 dates) to view the market trend. This will give a tabular view of currencies and its maximum profit with time on a daily basis.

Eg: Market trend of Currency ‘ETC’ from 1/6/2016 to 5/6/2016 on a daily basis

# Technologies Used

* React and Node Js for UI module which can be deployed in a container
* Springboot Micro service architecture in service layer which can also be deployed in a container
* Mongo DB as Repository
* Test Driven approach by integrating Junit test case classes

# Solution Architecture



*As part of implementation, I have created independent layers for UI and Service[logic]. Both of these layers can be handled independently.*

*Any change can be made in these modules easily without affecting other modules.*

*Test classes and methods are built in parallel to development to ensure code stability.*

*Test cases and data are prepared prior to implementation*

**UI Module**

* Front end components are built using React and it is a Single page application
* NodeJS is acting as the backend service provider for UI Module
* Used Native HTML tags embedded with react component (jsx)
* implemented Ajax calls to communicate with service layer
* This module can be scaled/changed separately without affecting other layers

**Service Module**

* Service layer is built using spring boot micro service architecture
* Currently, it can be executed as a standalone Rest service
* Rest APIs have been implemented to receive inputs from UI
* Mongo DB driver has been used to communicate with Database
* This can be extended to proper MS architecture using Service registry, gateway and load balancer / using kubernetes.

**DB Layer**

* Mongo DB is the repository that used in this approach
* It stores data in json format and relieves the same from service layer using driver integration
* All details are entered in DB prior to application Testing

# Details design of Logic for identifying profit

I have implemented this through double iteration after sorting using time. Each price will be compared with all future prices to identify maximum profit

Custom comparator is implemented for simplifying future details identification.

**Algorithm**

Declare and initialize max difference;

Declare variable to store currency details

Iterate through quotes

for each currency

Iterate though future prices

If (difference>maxDeffernce)

Set max difference to new value and save currency details

End If

End loop

End loop

# Test Details & Screenshot

* The application is implemented in a Test driven method where test data and cases were prepared prior to development.
* Test classes are created in parallel to development to ensure both REST API and Logic are never broken
* Test classes are implemented using JUNIT in springboot

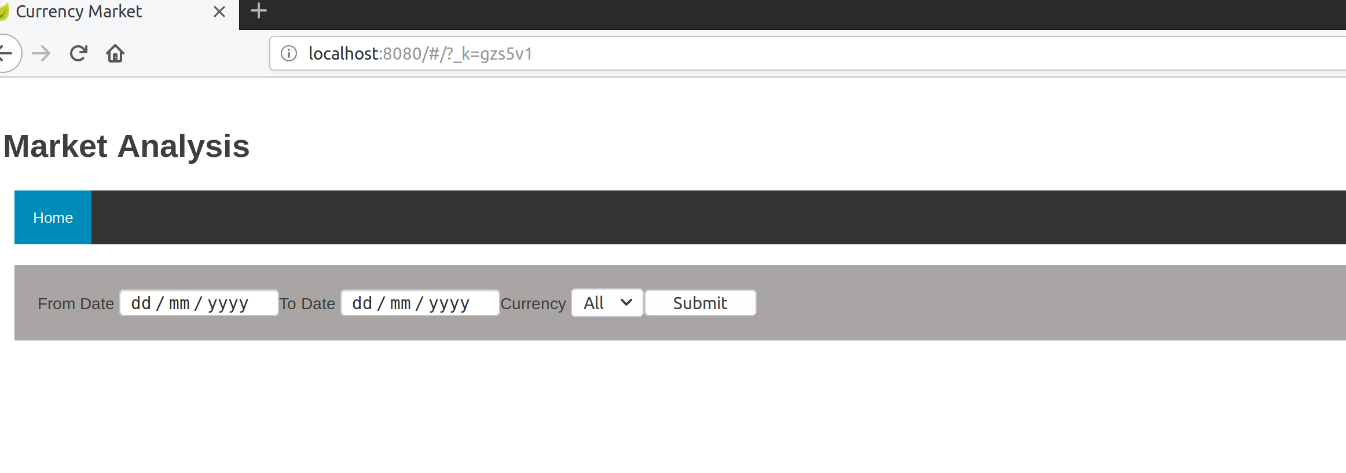
**Post Deployment Test Data**

Test Case 1:

Scenario: User should be able to see User interface on clicking URL and he should be able to edit the fields

Expected Result: UI with required fields are populated

Actual Result: Screenshot

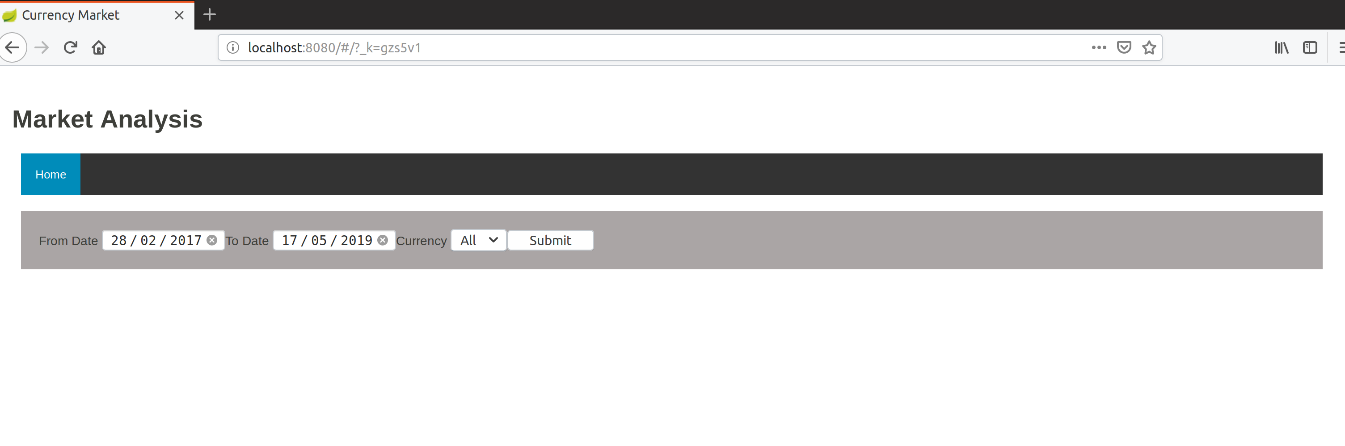


Test Case 2:

Scenario: User should be able to edit fields and click on submit

Expected Result: UI allows user to edit from and to date and currency

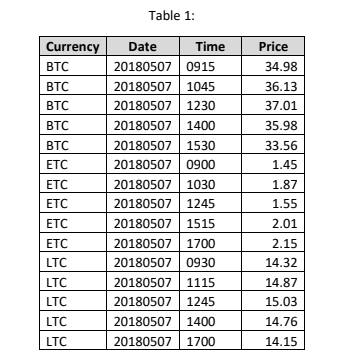
Actual Result: Screenshot



Test Case 3:

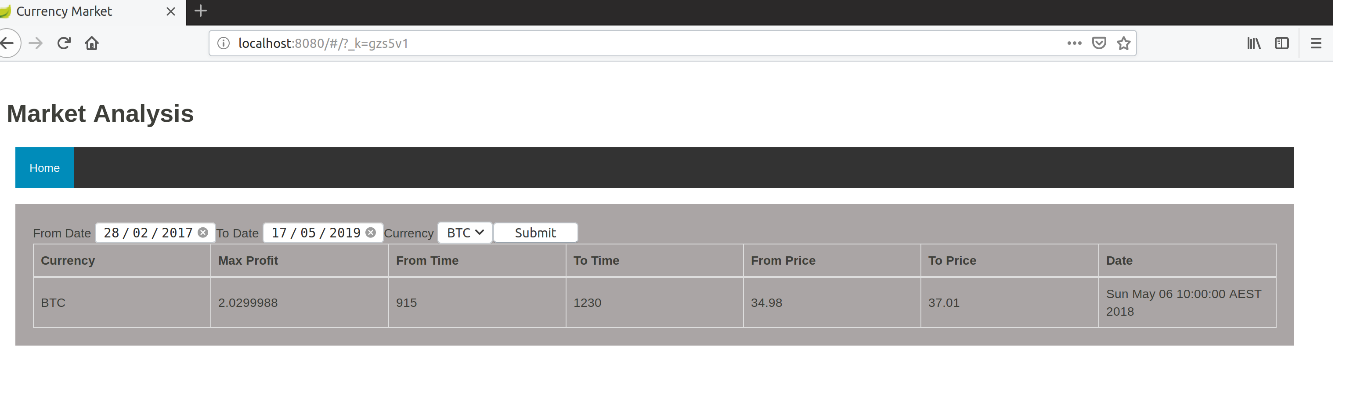
Scenario: Get Maximum profit and time for ‘BTC’ currency

Test Data:



Expected Result: UI with from time as 9:12 and to time as 12:30

Actual Result: Screenshot



Test Case 4:

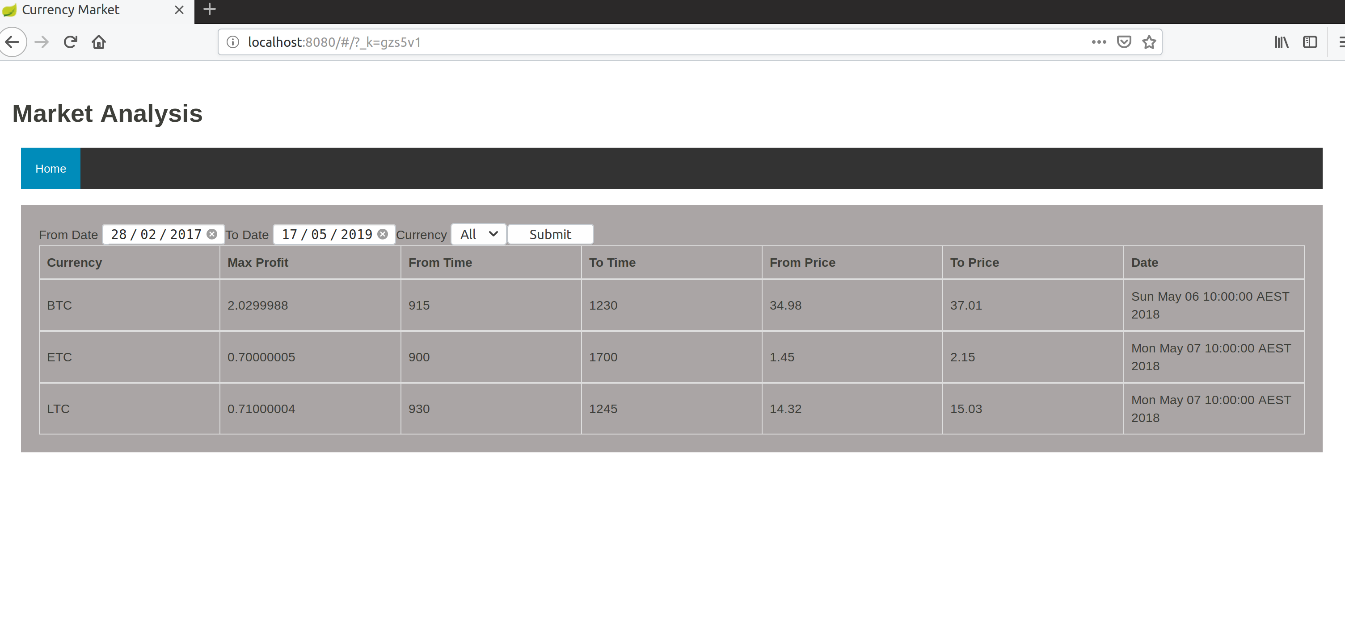
Scenario: Get Maximum profit and time for all currencies which includes currency submitted date

Expected Result: Expected Result: UI with from time as 9:00 and to time as 17:00 for ETC

And from time as 9:30 and to time as 12:45 for LTC

And from time as 9:15 and to time as 12:30 for BTC

Actual Result: Screenshot



# Enhancement

* UI module can be enhanced with Bootstrap integration
* Apply Containerisation with Docker
* Add CI/CD pipelines

.

# References

|  |  |  |  |
| --- | --- | --- | --- |
| SI. No. | Document Number | Title | Version |
| 1 |  | Internet blogs and study materials |  |
|  |  |  |  |

# Definitions

The following words, acronyms and abbreviations are referred to in this document.

| Term | Definition |
| --- | --- |
| GUI | Graphical User Interface |
| REST | Representational State Transfer |
| CI/CD | Continuous Integration/Continuous Deployment |

# Attachments

|  |  |  |
| --- | --- | --- |
| SI. No. | Title | Document |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Documentation location

|  |  |
| --- | --- |
| Master Hard copy | Master Electronic |
| NA |  |

# DOCUMENT CONTROL SHEET

The purpose of this section is to capture all changes made to the content of document.

Contact for Enquiries and Proposed Changes

If you have any questions regarding this document or would like to suggest an improvement, contact:

|  |  |
| --- | --- |
| Name | Varun K R |
| Designation | Technology Lead |
| Phone |  |
| Fax |  |