**Online Calculator System Design Document**

Table of Contents

[Purpose 3](#_Toc85443877)

[Problem 3](#_Toc85443878)

[Scope 3](#_Toc85443879)

[Assumptions 3](#_Toc85443880)

[Design Diagram 4](#_Toc85443881)

[Class Diagram 4](#_Toc85443882)

[Code Map Diagram 4](#_Toc85443883)

[Sequence Diagram 5](#_Toc85443884)

[Implementation Details 6](#_Toc85443885)

[Approach 6](#_Toc85443886)

[Code 6](#_Toc85443887)

[Extensibility 7](#_Toc85443888)

[Deployment 7](#_Toc85443889)

[Testing 7](#_Toc85443890)

[Unit Tests 7](#_Toc85443891)

[Manual Verification 8](#_Toc85443892)

[Reliability 9](#_Toc85443893)

[Scalability 9](#_Toc85443894)

[Workload / Concurrency 9](#_Toc85443895)

[Security 9](#_Toc85443896)

[Multiple Regions 9](#_Toc85443897)

[Limitations 9](#_Toc85443898)

# Purpose

The purpose of this document is to describe system design and implementation details for Online Calculator.

# Problem

Provide system design for an Online Calculator. The Online Calculator should support:

1. Parsing complex arithmetic expressions e.g. (4+2) x 4 / 2

2. Should support Memory Save and Recall e.g. <MR> + 3

3. Bonus: Memory Save should persist across sessions

         4. Bonus: Memory save for one user should not be visible to other users

# Scope

To provide design and implementation details on online calculator that can evaluate a simple infix expression and provides basic support for session and memory management.

# Assumptions

Following assumptions are considered while designing and implementing the app.

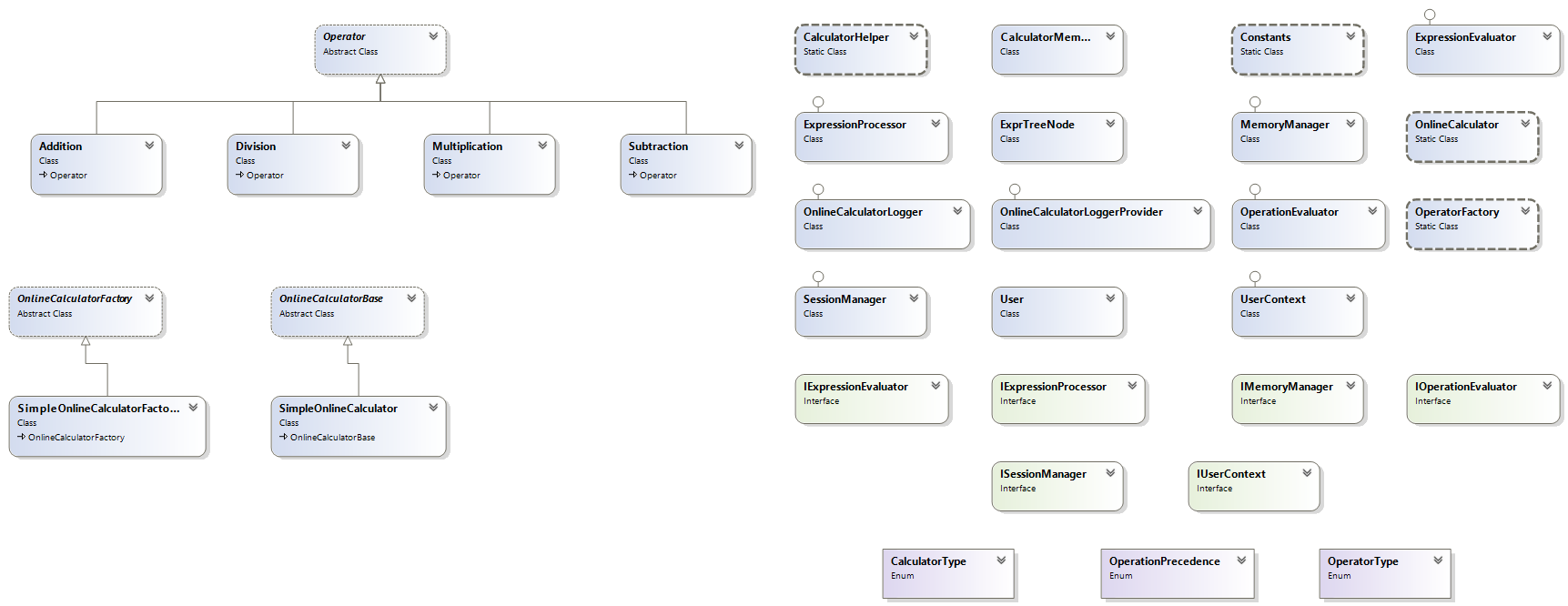
* Currently, only supports arithmetic operations on integers.
* Currently, only basic arithmetic operations (+, -, \* and /).
* Result of the expression will be saved after evaluation.
* Memory recall will retrieve the result of previously evaluated expression or 0.
* Although there are expression sanitization and basic validations are present. Perfect input expression would be expected.
* Supports only basic session management.
* User authentication is supported only by function keys.
* Supports only basic telemetry to indicate function start and end events.
* Supports single letter operations. (+, -, \*, / and R (Memory recall))

# Design Diagram

## Class Diagram

The below class diagram is generated using Visual Studio 2019 by Class Designer extension.

**Class diagram for Online Calculator App**



## Code Map Diagram

The following code map diagram shows the call interaction and dependency between different objects of the application.

Color indicator

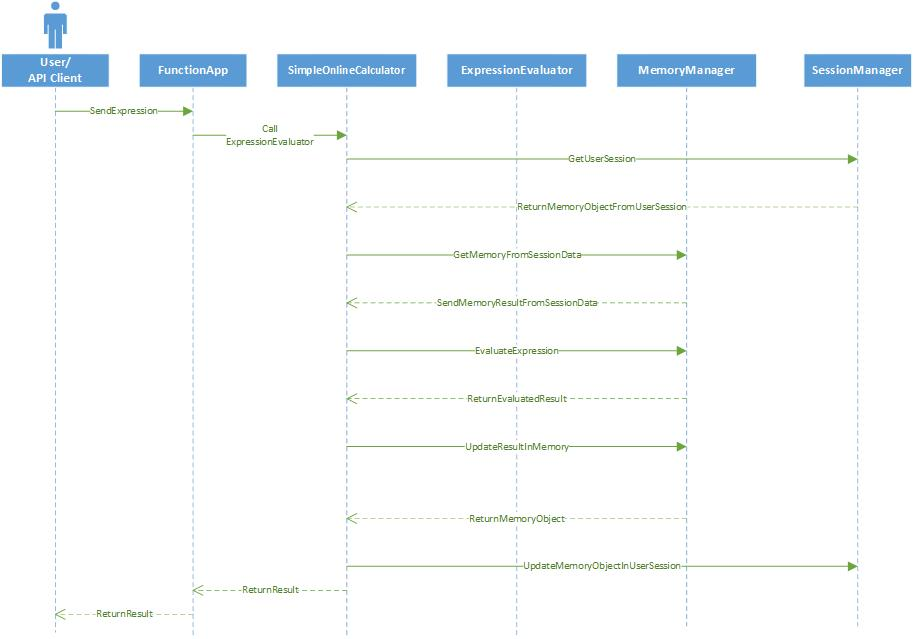
|  |  |
| --- | --- |
| Arrow Color | Category |
| Green | Inheritance |
| Pink | Calls |
| Grey | References |
| Dotted Green | Interface |

Diagram

Description automatically generated

## Sequence Diagram

The sequence of actions between the objects while evaluation expression is depicted in the below sequence diagram.



# Implementation Details

## Approach

Implemented as Azure Function App using .NET Core 3.0 in C# language. The application is implemented in extendable approach by following required software design patterns and SOLID principles.

This calculator supports the following features

1. Evaluates the arithmetic expressions with basic operations like Addition, Subtraction, Multiplication and Division.
2. Once the expression is evaluated the result will be stored in session of the user.
3. The result will be managed across session of the user.
4. The memory of one user will not be visible to another user.
5. The memory can be recalled using operator ‘**R**’.
6. The recollected memory can also be used in operations.

## Code

Code is deployed into a GitHub repository.

GitHub repo : [BalajiDabbara/mmd-test-projects (github.com)](https://github.com/BalajiDabbara/mmd-test-projects)

Follow the below steps to run the app

1. Clone the Git repo
2. Open Visual Studio 2019 in administration mode.
3. Open the solution file ***(\OnlineCalculator\Online Calculator App \Online Calculator App .sln***)
4. Build the solution
5. Run the App (Ctrl + F5)
6. Local function app will be started.
7. Test using Postman or any other API tester.

# Extensibility

This app is designed which can be extended further in the future. The following operations are not supported in the current implementation and can be extended.

* Implemented to be extended to different type of operators.
* Advanced calculator.
* Custom logger for telemetry logging.
* Unary operator
* Double digit operators.
* User level authentication.
* On-demand Save
* Multi-level Save.
* Dependence injection container.
* Durable function app.
* Memory clear.
* Advanced Session management.

# Deployment

The Online Calculator function app is deployed in Azure cloud as a Function App.

# Testing

Online Calculator App is tested by unit tests with xUnit framework and evaluated the API using Postman.

Added unit tests are added in the project ***Online Calculator App .Tests***

### Unit Tests

Added unit tests are added in the project ***Online Calculator App .Tests***

Following are the different test scenarios added for the function app.

1. ValidateInfixExpressionEvaluationSuccess
2. ValidateInfixExpressionEvaluationFailedOnInvalidInput
3. ValidateInfixExpressionEvaluationSuccesssOnAllOperators
4. ValidateEmptyUserNameThrowsAnError
5. ValidateMemoryRecall
6. ValidateOperationOnMemoryRecall
7. ValidateMemoryRecallForDifferentUsers
8. ValidateSupportForMaximumLongValue
9. ValidateSupportForMinimumLongValue
10. ValidateSupportSingleNegativeNumber

### Manual Verification

Online Calculator App API functionality can be validated using Postman. The URL and request body to be passed for the request should be as shown below.

Azure Deployed Function App API URL:

https://onlinecalculatorapp.azurewebsites.net/api/OnlineCalculator\_Evaluate

|  |  |
| --- | --- |
| Request URL | https://onlinecalculatorapp.azurewebsites.net/api/OnlineCalculator\_Evaluate |
| Content Type | *json* |
| Request Body | *{*  *"UserName": "Balaji",*  *"InfixExpression": "(((2+2)\*4/2+2))\*10/5\*2\*100"*  *}* |
| Response Body | Hello, Balaji. Your input expression ((((2+2)\*4/2+2))\*10/5\*2\*100) has been evaluated to : 4000. |

##### Postman

Graphical user interface, text, application, email

Description automatically generated

##### Azure App Verification

Graphical user interface, text, application, email

Description automatically generated

# Reliability

This depends on the consumption plan.

[Azure Functions best practices | Microsoft Docs](https://docs.microsoft.com/en-us/azure/azure-functions/functions-best-practices?tabs=csharp)

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-best-practices?tabs=csharp>

# Scalability

Can this be scaled?

Based on plan auto scale features will be available in Azure Function Apps

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-scale#timeout>

# Workload / Concurrency

Handling throttling scenarios and increasing workloads.

[Azure Functions scale and hosting | Microsoft Docs](https://docs.microsoft.com/en-us/azure/azure-functions/functions-scale#service-limits)

# Security

How to maintain the security of function app APIs?

1. By Authentication,
2. Function keys
   1. Function App level
   2. Function level

# Multiple Regions

How to maintain multiple regions?

By deploying in multiple regions. This may increase operational costs.

1. Can we maintain multiple regions with idempotent costs?
2. Will CI\CD server the purpose?
3. What about resources in multiple regions?

# Limitations

Online Calculator App has following limitations.

* Two or more letter operators are not supported (e.g.: ++, --, log, sin, cos).
* This is not extendable to double, float. Should be implemented with Generics.
* Can't support multiple types of braces are not supported like {, [ etc.
* Can't support multiple expression validations.
* User authentication process is not implemented
* Session State providers are not supported.
* Not validated for reliability.
* Not validated for workload.
* If input is only Int64.MinValue then this is not working.