Simple Calculator

SE452 – Object Oriented Enterprise Application Development

Sample project

Contents

[Team members 2](#_Toc352544209)

[Overview 2](#_Toc352544210)

[Requirements 2](#_Toc352544211)

[Use Case 2](#_Toc352544212)

[Description of problem 2](#_Toc352544213)

[Design 3](#_Toc352544214)

[Sequence of major functionality 3](#_Toc352544215)

[Web UI (Common case) 3](#_Toc352544216)

[Table layout 3](#_Toc352544217)

[Deployment 3](#_Toc352544218)

[Discussion of how your design met the requirements 3](#_Toc352544219)

[Discussion of lessons learned 4](#_Toc352544220)

[Decision Log 5](#_Toc352544221)

# Overview

Basic calculator (+, -, \*, /) functionality for human.

# Requirements

## Use Case



## Description of problem

Basic calculator operation service that supports human interaction using Web.

# Design

## Sequence of major functionality

### Web UI (Common case)



## Table layout



There is only one table to fulfill the requirement for persistence.

## Deployment

All the classes were packaged into one deployment unit to keep things simple.



Queue: CalculatorQ

Data source: jdbc/sample

# Discussion of how your design met the requirements

This sample project did not meet the functional requirements but was created to validate the component interaction for different technology pieces.

This also serves as a reference document for other projects of basic expectation of the course project work

# Discussion of lessons learned

# Decision Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem** | **What was decided** | **Alternatives considered** | **Rationale** |
| Which IDE to use | Netbean | Eclipse | Netbeans had all the components integrated in one UI |
| Collaboration | Mind mending | Course provided | We worked like one anyway |
| Code repo | Laptop | Course provided | We could afford only one powerful machine for this work |