Simple Graph Calculator and Plotter
- Developer Guide -

Norbert Jahn

July 27, 2025

Document Number	Revision 1.1
Responsible	Norbert Jahn
Status	Draft
Confidentiality	Internal

Contents

1 Introduction

This document contains the functional analysis and software design specifications of the ”Simple Graph Calculator and Plotter” application. This application is used as prototype of a software module used to calculate and display several mathematical functions on a graphical user interface such as sin, cos and sinc.

2 Revision history

Table 1: Document history

Revision	**Date**	**Author**	**Comment**
1.0	20.07.2025	Norbert Jahn	Creation of initial document
1.1	27.07.2025	Norbert Jahn	Added hierarchy image

3 Architecture Overview

This applications is structured based on the MVVM model. Figure 1 shows the four main parts and their relations.

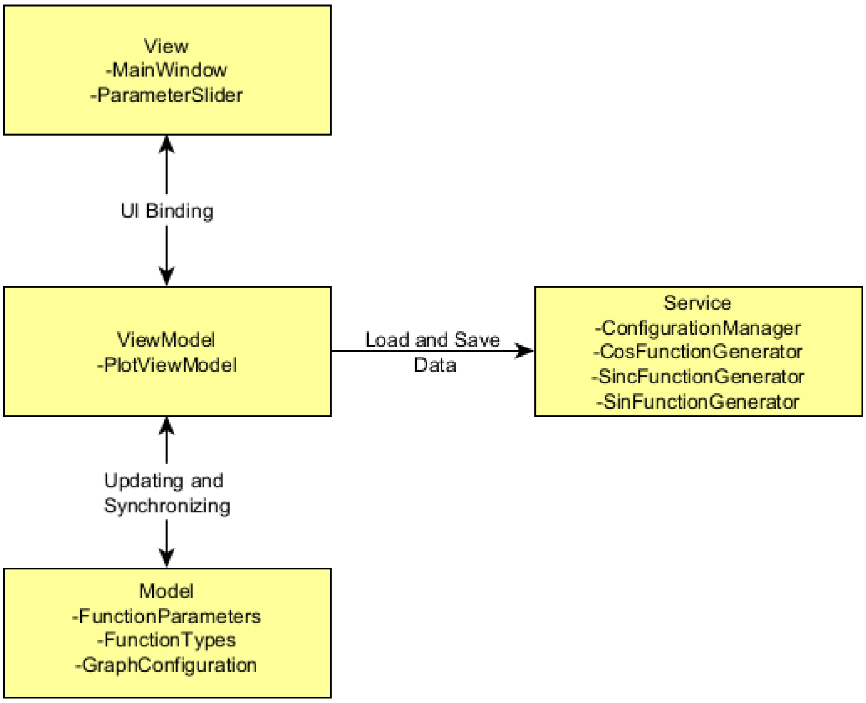

Figure 1: Software Structure

4 Folder Structure Explanation

This section describes the overall folder and namespace structure of the project, helping developers navigate and understand the codebase efficiently.

- **Model/**
  Contains data structures and business logic representations. This includes:
  - `FunctionParameters.cs`: Stores amplitude, frequency, phase, and axis ranges
  - `FunctionTypes.cs`: Enum defining available function types (Sin, Cos, Sinc)
  - `GraphConfiguration.cs`: Stores history for all graphs and last selected function
- **Properties/**
  Contains data general application data and tool settings. This includes:
  - `AssemblyInfo.cs`: Stores programm version, title, and company information
  - `Resources.Designer.cs`: Autogenerated file for resource settings
  - `Settings.Designer.cs`: Autogenerated file for desinger settings

- **Resources/**
  Contains custom logo and UI-Styles:
  - `lis_spg.ico`: Icon for this application
  - `Styles.xaml`: Containts specific styles for buttons and text used in the UI
- **Service/**
  Holds the mathematical function implementations used to compute graph points
  - `SinFunctionGenerator.cs`, `CosFunctionGenerator.cs`, `SincFunctionGenerator.cs`: Each implements the `IFunctionGenerator` interface.
- **View/**
  Contains the XAML views for the UI (WPF) except for the MainWindow. Each view is typically bound to a corresponding ViewModel
  - `ParameterSlider.xaml`, `ParameterSlider.xaml.cs`: Custom Slider for function parameters
- **ViewModel/**
  Implements the MVVM pattern. Contains background logic that binds UI elements to the data and handles plot generation
  - `PlotViewModel.cs`: Core class responsible for managing the plot, selected function, and parameters.

5 QA Test Considerations

This section contains testing strategies and scenarios for this prototype software. Also a list of known issues and error messages of this application is provided here.

5.1 Test Strategy

- Verify correct function rendering for edge values (e.g., large frequency)
- Ensure parameter are loaded correctly across multiple sessions
- Test export functionality with different file names and paths
- Validate user input: prevent empty or invalid numeric fields

5.2 Test Case List

- Leave function parameter fields empty
- Use characters instead of numbers in the function parameter fields
- Try to save graph configurations while invalid values are set in the function parameter fields
- Manipulate the Json file (app.json in Settings directory)
- Execute programm from folder that requires administrator privileges to write to
- Delete settings directory before launching the app
- Delete settings directory during runtime
- Try to save images in a directory with a very long path (more than 255 characters)
- Try to save images in a protected directory (System)

5.3 Error messages and meaning

Error Message	Explanation
You do not have permission to write to the selected location.	The specified file path is invalid. Make sure this application has sufficient privileges.
An unexpected error occurred, check if there is enough space on your hard drive	An unspecified exception occured during saving the image. A possibility for this is that there is not enough free space on the harddrive.
Amplitude must not be smaller than 0!	No negative values are allowed for amplitude. 0 is the lowest value the function accepts
Frequency must not be smaller than 0!	No negative values are allowed for frequency. 0 is the lowest value the function accepts
Phase must not be smaller than 0!	No negative values are allowed for phase. 0 is the lowest value the function accepts

Table 2: Common error messages and their explanations

5.4 Errata Sheet

- Functions are not displayed correctly when high frequencies set
- The software crashes when the "Settings" directory which contains the app.json is deleted
- The textboxes for x-min and x-max are still accepting letters instead of numerical values

6 Required Packages

The application depends on the following NuGet packages for compilation and testing:

Package Name	Version	Target Framework
Microsoft.NETFramework .ReferenceAssemblies	1.0.3	net48
Microsoft.NETFramework .ReferenceAssemblies.net48	1.0.3	net48
OxyPlot.Core	2.2.0	net48
OxyPlot.Wpf	2.2.0	net48
OxyPlot.Wpf.Shared	2.2.0	net48
System.Buffers	4.6.1	net48
System.Memory	4.6.3	net48
System.Numerics.Vectors	4.6.1	net48
System.Runtime. CompilerServices.Unsafe	6.1.2	net48

System.Threading.Tasks. Extensions	4.6.3	net48