

SE 3XA3: Module Guide Ultimate Calculator

Group 15 L01
Jarod Rankin, rankij5
Mathew Petronilho, petronim
Logan Brown, brownl33
Syed Bokhari, bokhars

April 8, 2022

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Scope	1
1.3	Acronyms, Abbreviations, and Symbols	1
1.4	Overview of Document	2
2	Anticipated and Unlikely Changes	2
2.1	Anticipated Changes	2
2.2	Unlikely Changes	3
3	Module Hierarchy	4
4	Connection Between Requirements and Design	6
5	Module Decomposition	7
5.1	Hardware Hiding Modules	7
5.2	Behaviour-Hiding Modules	7
5.2.1	Main Calculator GUI Module (M1)	7
5.2.2	Algebra GUI Module (M3)	7
5.2.3	Slope GUI Module (M5)	7
5.2.4	Y-intercept GUI Module (M6)	8
5.2.5	Pythagorean Theorem GUI Module (M7)	8
5.2.6	Health GUI Module (M8)	8
5.2.7	BMI GUI Module (M10)	8
5.2.8	Body Fat GUI Module (M11)	8
5.2.9	Conversion GUI Module (M12)	9
5.2.10	Currency Conversion GUI Module (M14)	9
5.2.11	Crypto Conversion GUI Module (M15)	9
5.2.12	Base Conversion GUI Module (M16)	9
5.2.13	Roman Numerals Conversion GUI Module (M17)	9
5.2.14	Geometry GUI Module (M18)	10
5.2.15	Area GUI Module (M20)	10
5.2.16	Perimeter GUI Module (M21)	10
5.2.17	Volume GUI Module (M22)	10
5.2.18	Binary GUI Module (M23)	10
5.2.19	Floating Point GUI Module (M25)	11
5.2.20	Floating Point GUI Module (M26)	11
5.2.21	Bitwise Operation GUI Module (M27)	11
5.2.22	Stocks GUI Module (M28)	11
5.2.23	GPA GUI Module (M30)	11
5.3	Software Decision Module	12
5.3.1	Main Calculator Module (M2)	12
5.3.2	Algebra Calculator Module (M4)	12

5.3.3	Health Calculator Module (M9)	12
5.3.4	Roman Numerals Conversion Module (M13)	12
5.3.5	Geometry Calculator Module (M19)	13
5.3.6	Binary Calculator Module (M24)	13
5.3.7	Stocks Calculator Module (M29)	13
5.3.8	GPA Calculator Module (M31)	13
6	Traceability Matrix	13
7	Use Hierarchy Between Modules	15

List of Tables

1	Revision History	ii
2	Table of Abbreviations	1
3	Table of Definitions	2
4	Module Hierarchy	6
5	Trace Between Requirements and Modules	14
6	Trace Between Anticipated Changes and Modules	15

List of Figures

1	Use hierarchy among modules	17
---	-----------------------------	----

Table 1: **Revision History**

Date	Version	Notes
March 15, 2022	1.0	Modules added to Module Hierarchy
March 16, 2022	1.1	Introduction Added
March 16, 2022	1.2	Anticipated and Unlikely Changes Added, some Module Decompositions Added
March 18, 2022	1.3	Module Decompositions completed
March 18, 2022	1.4	Uses Hierarchy and Traceability Matrix Added
April 7, 2022	1.5	Updated Module Guide for Revision 1 submission

1 Introduction

The Ultimate Calculator is an open source calculator, written in python. The goal of this project is to upgrade the GUI of the current project, and add additional functionality to the calculator. These additions will be implemented using python and PyQt5. Additionally, these features will be thoroughly tested to ensure they are as good as they can be.

1.1 Purpose

The purpose of this document is to show the modular structure of our system. The system should allow for project maintainers and designers to be able to follow this document and clearly see how the parts of the software are modularized, giving the system high maintainability. This document shall show the focus of maintaining high coupling and low cohesion between the systems modules

1.2 Scope

The module guide follows the requirements found in the SRS document in the GitLab repository. The modules found in the module guide have their functionality described in the Module Interface Specification(MIS) found in the projects GitLab repository. This document will outline the hierarchy of the modules of the system and will outline the anticipated and unlikely changes of the system.

1.3 Acronyms, Abbreviations, and Symbols

Table 2: **Table of Abbreviations**

Abbreviation	Definition
GUI	Graphical User Interface
SRS	Software Requirements Specification
FR	Functional Requirement
NFR	Non-Functional Requirement
GPA	Grade Point Average
OS	Operating System
BMI	Body Mass Index

Table 3: **Table of Definitions**

Term	Definition
Operation	Any mathematical function that takes in one or more parameters and outputs a well-defined answer
Operation Type	Class of operations with common characteristics
Operation Section	A window that relates to a specified operation and displays the necessary parameters and result for that operation (includes the main menu)
Computation	Finding the answer to a problem via mathematics
Python	The programming language used to develop Ultimate Calculator
User	The individual interacting with the application
Offline	Accessing the application without the use of an internet connection
PyQt5	GUI toolkit used for Ultimate Calculator
Window	Separate area of the display of Ultimate Calculator
Input Parameters	The area where the user inputs values for calculations

1.4 Overview of Document

The rest of the document is outlined as so:

- Section 2 outlines any anticipated and unlikely changes to the systems requirements.
- Section 3 outlines the hierarchy of the systems modules while taking into consideration the likely changes to the requirements.
- Section 4 outlines the connections between the systems specified requirements and the systems modules.
- Section 5 outlines the description of each module vividly.
- Section 6 outlines the traceability matrices of the system. These matrices relate to the requirements found in the SRS and the likely changes of the systems modules.
- Section 7 outlines how each modules interacts with one another.

2 Anticipated and Unlikely Changes

This section lists possible changes to the system. According to the likeliness of the change, the possible changes are classified into two categories. Anticipated changes are listed in Section 2.1, and unlikely changes are listed in Section 2.2.

2.1 Anticipated Changes

Anticipated changes are the source of the information that is to be hidden inside the modules. Ideally, changing one of the anticipated changes will only require changing the one module that hides the associated decision. The approach adapted here is called design for change.

- AC1:** A new operation type is added (~~main module UI~~).
- AC2:** A new converter is added (~~converter menu UI~~).
- AC3:** A new algebraic operation is added (~~algebra menu UI~~).
- AC4:** A new stock operation is added (~~stock menu UI~~).
- AC5:** A new GPA calculator based on a different scale is added (~~GPA menu UI~~).
- AC6:** A new geometric operation is added (~~geometry menu UI~~).
- AC7:** A new binary operation is added (~~binary menu UI~~).
- AC8:** A new health operation is added (~~health menu UI~~).
- AC9:** A change in the algorithm for algebraic operations (~~Algebra functionality module~~).
- AC10:** A change in the algorithm for GPA operations (~~GPA functionality~~).
- AC11:** A change in the algorithm for geometric operations (~~geometric functionality~~).
- AC12:** A change in the algorithm for binary operations (~~binary functionality~~).
- AC13:** A change in the algorithm for health operations (~~health functionality~~).

2.2 Unlikely Changes

The module design should be as general as possible. However, a general system is more complex. Sometimes this complexity is not necessary. Fixing some design decisions at the system architecture stage can simplify the software design. If these decision should later need to be changed, then many parts of the design will potentially need to be modified. Hence, it is not intended that these decisions will be changed.

- UC1:** Changes in Input/Output devices such as mouse, keyboard and display.
- UC2:** There will always be a source of input data external to the software.
- UC3:** Change in programming language as the application is highly dependant on Python and PyQt5.
- UC4:** Changes in OS as the application is built specifically for MacOS and Windows.
- UC5:** Change to the design of the GUIs in the application; this would require a revamp of the entire application.
- UC6:** ~~Changes related to the use of the internet within the application.~~

3 Module Hierarchy

This section provides an overview of the module design. Modules are summarized in a hierarchy decomposed by secrets in Table 4. The modules listed below are the modules that will actually be implemented.

- M1:** Main Calculator GUI Module
- M2:** Main Calculator Module
- M3:** Algebra GUI Module
- M4:** Algebra Calculator Module
- M5:** Slope GUI Module
- M6:** Y-intercept GUI Module
- M7:** Pythagorean Theorem GUI Module
- M8:** Health GUI Module
- M9:** Health Calculator Module
- M10:** BMI GUI Module
- M11:** Body Fat GUI Module
- M12:** Conversion GUI Module
- M13:** Conversion Module
- M14:** Currency Conversion GUI Module
- M15:** Crypto Conversion GUI Module
- M16:** Base Conversion GUI Module
- M17:** Roman Numerals Conversion GUI Module
- M18:** Geometry GUI Module
- M19:** Geometry Module
- M20:** Area GUI Module
- M21:** Perimeter GUI Module
- M22:** Volume GUI Module
- M23:** Binary GUI Module
- M24:** Binary Calculator Module

- M25:** Floating Point GUI Module
- M26:** Binary Arithmetic GUI Module
- M27:** Bitwise Operation GUI Module
- M28:** Stocks GUI Module
- M29:** Stocks Calculator Module
- M30:** GPA GUI Module
- M31:** GPA Calculator Module

Level 1	Level 2
Hardware-Hiding Module	N/A
Behaviour-Hiding Module	Main Calculator GUI Module Algebra GUI Module Slope GUI Module Y-intercept GUI Module Pythagorean Theorem GUI Module Health GUI Module BMI GUI Module Body Fat GUI Module Conversion GUI Module Currency Conversion GUI Module Crypto Conversion GUI Module Base Conversion GUI Module Roman Numerals Conversion GUI Module Stocks GUI Module GPA GUI Module Geometry GUI Module Area GUI Module Perimeter GUI Module Volume GUI Module Binary GUI Module Floating Point GUI Module Binary Arithmetic GUI Module Bitwise Operation Module
Software Decision Module	Main Calculator Module Algebra Calculator Module Health Calculator Module Conversion Module Stocks Calculator Module GPA Calculator Module Geometry Module Binary Calculator Module

Table 4: Module Hierarchy

4 Connection Between Requirements and Design

The system of **Ultimate Calculator** is intended to ensure that all of the functional and non-functional requirements developed in the SRS are incorporated into the design. The application design is modularized with functionalities being clear indicators to varying requirements. The network of connections between the requirements and modules can be seen via the traceability matrices in Table 5.

5 Module Decomposition

Modules are decomposed according to the principle of “information hiding” proposed by Parnas et al. (1984). The *Secrets* field in a module decomposition is a brief statement of the design decision hidden by the module. The *Services* field specifies *what* the module will do without documenting *how* to do it. For each module, a suggestion for the implementing software is given under the *Implemented By* title.

5.1 Hardware Hiding Modules

N/A

5.2 Behaviour-Hiding Modules

Secrets: The contents of the required behaviours.

Services: Includes programs that provide externally visible behaviour of the system as specified in the software requirements specification (SRS) documents. This module serves as a communication layer between the hardware-hiding module and the software decision module. The programs in this module will need to change if there are changes in the SRS.

Implemented By: –

5.2.1 Main Calculator GUI Module (M1)

Secrets: The design format of the Main Calculator GUI.

Services: Creates the window for the Main Calculator with clickable buttons that allow the user to navigate the application.

Implemented By: Ultimate Calculator

5.2.2 Algebra GUI Module (M3)

Secrets: The design format of the Algebra GUI.

Services: Creates the window for the Algebra operation type with clickable buttons that allow the user to navigate the application.

Implemented By: Ultimate Calculator

5.2.3 Slope GUI Module (M5)

Secrets: The design format of the Slope GUI.

Services: Creates the window for the Slope operation with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.4 Y-intercept GUI Module (M6)

Secrets: The design format of the Y-intercept GUI.

Services: Creates the window for the Y-intercept operation with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.5 Pythagorean Theorem GUI Module (M7)

Secrets: The design format of the Pythagorean Theorem GUI.

Services: Creates the window for the Pythagorean Theorem operation with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.6 Health GUI Module (M8)

Secrets: The design format of the Health GUI.

Services: Creates the window for the Health operation type with clickable buttons that allow the user to navigate the application.

Implemented By: Ultimate Calculator

5.2.7 BMI GUI Module (M10)

Secrets: The design format of the BMI GUI.

Services: Creates the window for the BMI operation with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.8 Body Fat GUI Module (M11)

Secrets: The design format of the Body Fat GUI.

Services: Creates the window for the Body Fat operation with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.9 Conversion GUI Module (M12)

Secrets: The design format of the Conversion GUI.

Services: Creates the window for the Conversion operation type with clickable buttons that allow the user to navigate the application

Implemented By: Ultimate Calculator

5.2.10 Currency Conversion GUI Module (M14)

Secrets: The design format of the Currency Conversion GUI.

Services: Creates the window for the Currency Conversion operation with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.11 Crypto Conversion GUI Module (M15)

Secrets: The design format of the Crypto Conversion GUI.

Services: Creates the window for the Crypto Conversion operation with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.12 Base Conversion GUI Module (M16)

Secrets: The design format of the Base Conversion GUI.

Services: Creates the window for the Base Conversion operation with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.13 Roman Numerals Conversion GUI Module (M17)

Secrets: The design format of the Roman Numerals Conversion GUI.

Services: Creates the window for the Roman Numerals Conversion operation with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.14 Geometry GUI Module (M18)

Secrets: The design format of the Geometry GUI.

Services: Creates the window for the Geometry operation type with clickable buttons that allow the user to navigate the application.

Implemented By: Ultimate Calculator

5.2.15 Area GUI Module (M20)

Secrets: The design format of the Area GUI.

Services: Creates the window for the Geometry operation with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.16 Perimeter GUI Module (M21)

Secrets: The design format of the Perimeter GUI.

Services: Creates the window for the Geometry operation with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.17 Volume GUI Module (M22)

Secrets: The design format of the Volume GUI.

Services: Creates the window for the Geometry operation with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.18 Binary GUI Module (M23)

Secrets: The design format of the Binary GUI.

Services: Creates the window for the Binary operation type with clickable buttons that allow the user to navigate the application.

Implemented By: Ultimate Calculator

5.2.19 Floating Point GUI Module (M25)

Secrets: The design format of the Floating Point GUI.

Services: Creates the window for the Binary operation type with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.20 Floating Point GUI Module (M26)

Secrets: The design format of the Binary Arithmetic GUI.

Services: Creates the window for the Binary operation type with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.21 Bitwise Operation GUI Module (M27)

Secrets: The design format of the Bitwise Operation GUI.

Services: Creates the window for the Binary operation type with clickable buttons and input fields that allow the user to enter parameters and obtain an answer.

Implemented By: Ultimate Calculator

5.2.22 Stocks GUI Module (M28)

Secrets: The design format of the Stocks GUI.

Services: Creates the window for the Stocks operation with clickable buttons and input fields that allow the user to enter the desired parameters and obtain a specific answer.

Implemented By: Ultimate Calculator

5.2.23 GPA GUI Module (M30)

Secrets: The design format of the GPA GUI.

Services: Creates the window for the GPA operation with clickable buttons and input fields that allow the user to enter the desired parameters and obtain a specific answer.

Implemented By: Ultimate Calculator

5.3 Software Decision Module

Secrets: The design decision based on mathematical theorems, physical facts, or programming considerations. The secrets of this module are *not* described in the SRS.

Services: Includes data structure and algorithms used in the system that do not provide direct interaction with the user.

Implemented By: –

5.3.1 Main Calculator Module (M2)

Secrets: The algorithms required for Main Calculator calculations.

Services: Provides the methods and functions necessary for addition, subtraction, multiplication, division, exponents and clearing the calculator.

Implemented By: Ultimate Calculator

5.3.2 Algebra Calculator Module (M4)

Secrets: The algorithms required for Algebraic calculations.

Services: Provides the methods and functions necessary to calculate slope, y-intercept, and the Pythagorean theorem.

Implemented By: Ultimate Calculator

5.3.3 Health Calculator Module (M9)

Secrets: The algorithms required for Health calculations.

Services: Provides the methods and functions necessary to calculate BMI and body fat.

Implemented By: Ultimate Calculator

5.3.4 Roman Numerals Conversion Module (M13)

Secrets: The algorithms required for currency, crypto, base and roman numeral conversion calculations.

Services: Provides the methods and functions necessary to calculate the currency, crypto, base and roman numeral conversions.

Implemented By: Ultimate Calculator

5.3.5 Geometry Calculator Module (M19)

Secrets: The algorithms required for Geometric calculations.

Services: Provides the methods and functions necessary to calculate area, perimeter, and volume of various shapes.

Implemented By: Ultimate Calculator

5.3.6 Binary Calculator Module (M24)

Secrets: The algorithms required for Binary calculations.

Services: Provides the methods and functions necessary to calculate floating point representation, binary arithmetic, and bitwise operations.

Implemented By: Ultimate Calculator

5.3.7 Stocks Calculator Module (M29)

Secrets: The algorithms required for the Stocks calculations.

Services: Provides the methods and functions necessary to calculate profit gain/loss.

Implemented By: Ultimate Calculator

5.3.8 GPA Calculator Module (M31)

Secrets: The algorithms required for the GPA calculations.

Services: Provides the methods and functions necessary to calculate ones GPA.

Implemented By: Ultimate Calculator

6 Traceability Matrix

This section shows two traceability matrices: between the modules and the requirements and between the modules and the anticipated changes.

Req.	Modules
FR1	M1
FR2	M1
FR3	M5, M6, M7, M10, M11, M14, M15, M16, M17, M20, M21, M22, M25, M26, M27, M28, M30
FR4	M1
FR5	M3, M5, M6, M7, M8, M10, M11, M12, M14, M15, M16, M17, M18, M20, M21, M22, M23, M25, M26, M27, M28, M30
FR6	M5, M6, M7, M10, M11, M14, M15, M16, M17, M20, M21, M22, M25, M26, M27, M28, M30
FR7	M5, M6, M7, M10, M11, M14, M15, M16, M17, M20, M21, M22, M25, M26, M27, M28, M30
FR8	M5, M6, M7, M10, M11, M14, M15, M16, M17, M20, M21, M22, M25, M26, M27, M28, M30
FR9	M5, M6, M7, M10, M11, M14, M15, M16, M17, M20, M21, M22, M25, M26, M27, M28, M30
FR10	M5, M6, M7, M10, M11, M14, M15, M16, M17, M20, M21, M22, M25, M26, M27, M28, M30
FR11	M5, M6, M7, M10, M11, M14, M15, M16, M17, M20, M21, M22, M25, M26, M27, M28, M30
FR12	M5, M6, M7, M10, M11, M14, M15, M16, M17, M20, M21, M22, M25, M26, M27, M28, M30
FR13	M5, M6, M7, M10, M11, M14, M15, M16, M17, M20, M21, M22, M25, M26, M27, M28, M30
FR14	M2, M4, M9, M13, M19, M24, M29, M31
FR15	M5, M6, M7, M10, M11, M14, M15, M16, M17, M20, M21, M22, M25, M26, M27, M28, M30
FR16	M5, M6, M7, M10, M11, M14, M15, M16, M17, M20, M21, M22, M25, M26, M27, M28, M30
FR17	M5, M6, M7, M10, M11, M14, M15, M16, M17, M20, M21, M22, M25, M26, M27, M28, M30
FR18	M5, M6, M7, M10, M11, M14, M15, M16, M17, M20, M21, M22, M25, M26, M27, M28, M30
FR19	M1
FR20	M1, M3, M5, M6, M7, M8, M10, M11, M12, M14, M15, M16, M17, M18, M20, M21, M22, M23, M25, M26, M27, M28, M30
FR21	M1

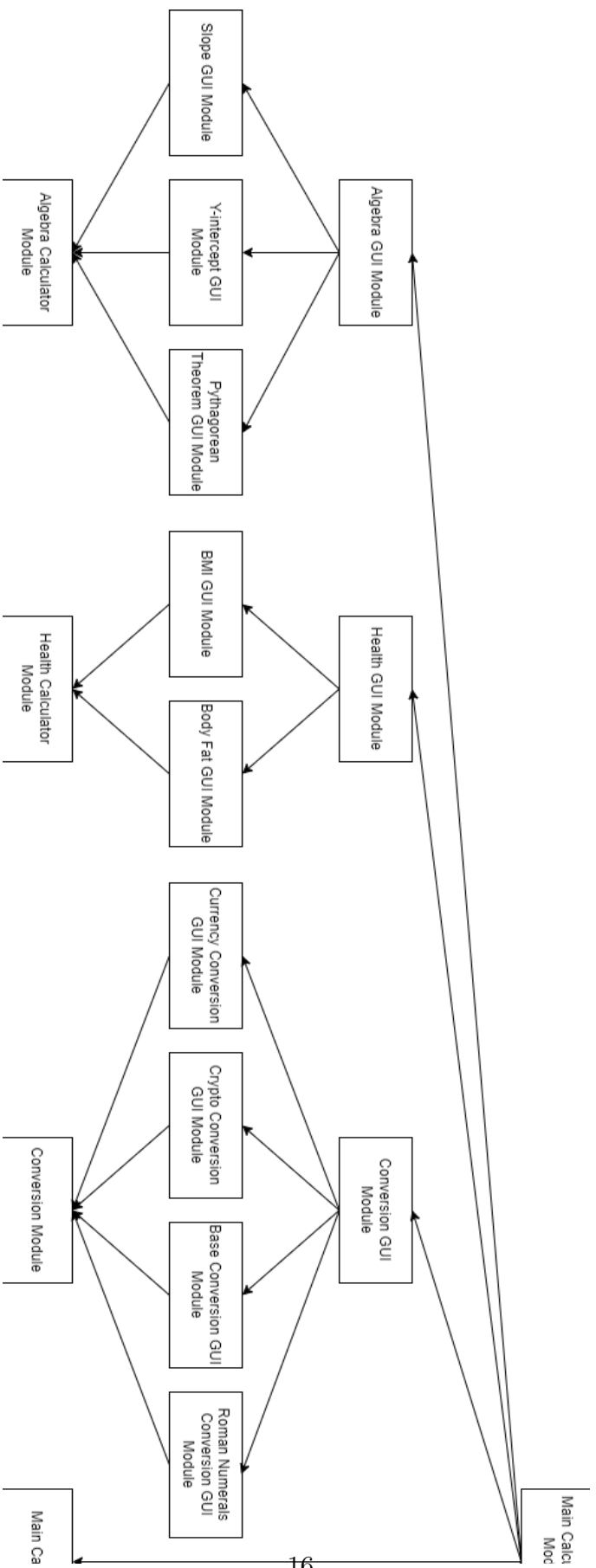
Table 5: Trace Between Requirements and Modules

AC	Modules
AC1	M1
AC2	M12
AC3	M3
AC4	M28
AC5	M30
AC6	M18
AC7	M23
AC8	M8
AC9	M4
AC10	M31
AC11	M19
AC12	M24
AC13	M9

Table 6: Trace Between Anticipated Changes and Modules

7 Use Hierarchy Between Modules

In this section, the uses hierarchy between modules is provided. Parnas (1978) said of two programs A and B that A *uses* B if correct execution of B may be necessary for A to complete the task described in its specification. That is, A *uses* B if there exist situations in which the correct functioning of A depends upon the availability of a correct implementation of B. Figure 1 illustrates the use relation between the modules. It can be seen that the graph is a directed acyclic graph (DAG). Each level of the hierarchy offers a testable and usable subset of the system, and modules in the higher level of the hierarchy are essentially simpler because they use modules from the lower levels.



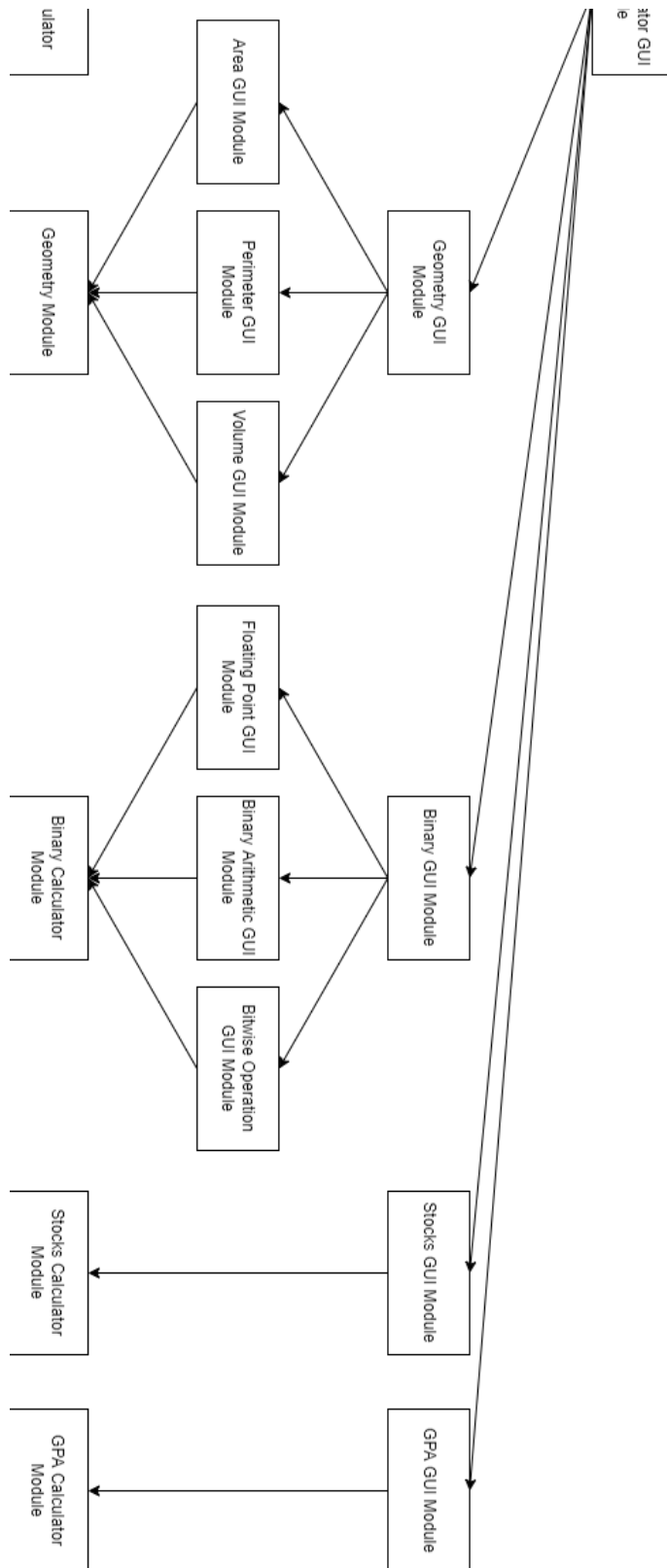


Figure 1: Use hierarchy among modules

References

- David L. Parnas. Designing software for ease of extension and contraction. In *ICSE '78: Proceedings of the 3rd international conference on Software engineering*, pages 264–277, Piscataway, NJ, USA, 1978. IEEE Press. ISBN none.
- D.L. Parnas, P.C. Clement, and D. M. Weiss. The modular structure of complex systems. In *International Conference on Software Engineering*, pages 408–419, 1984.