

SE 3XA3: Software Requirements
Specification
Ultimate Calculator

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

April 8, 2022

Contents

1	Project Drivers	1
1.1	The Purpose of the Project	1
1.2	The Stakeholders	1
1.2.1	The Client	1
1.2.2	The Customers	1
1.2.3	Other Stakeholders	1
1.3	Mandated Constraints	2
1.3.1	Solution Design Constraints	2
1.3.2	Partner or Collaborative Applications	2
1.3.3	Off-the-Shelf Software	2
1.3.4	Anticipated Workplace Environment	2
1.3.5	Schedule Constraints	3
1.3.6	Budget Constraints	3
1.3.7	Enterprise Constraints	3
1.4	Naming Conventions and Terminology	3
1.5	Relevant Facts and Assumptions	3
1.5.1	Facts	3
1.5.2	Assumptions	4
2	Functional Requirements	5
2.1	The Scope of the Work and the Product	5
2.1.1	The Context of the Work	5
2.1.2	Work Partitioning	5
2.1.3	Individual Product Use Cases	7
2.2	Functional Requirements	8
3	Non-functional Requirements	11
3.1	Look and Feel Requirements	11
3.2	Usability and Humanity Requirements	11
3.3	Performance Requirements	12
3.4	Operational and Environmental Requirements	13
3.5	Maintainability and Support Requirements	13
3.6	Security Requirements	14
3.7	Cultural Requirements	14
3.8	Legal Requirements	14
3.9	Health and Safety Requirements	14

4	Project Issues	14
4.1	Open Issues	14
4.2	Off-the-Shelf Solutions	14
4.3	New Problems	15
4.4	Tasks	15
4.5	Migration to the New Product	15
4.6	Risks	15
4.7	Costs	16
4.8	User Documentation and Training	16
4.8.1	User Documentation	16
4.8.2	Training	16
4.9	Waiting Room	16
4.10	Ideas for Solutions	16
5	Appendix	17
5.1	Symbolic Parameters	17

List of Tables

1	Revision History	iii
2	Naming Conventions and Terminology	3
3	Work Partitioning Events	5
4	Work Partitioning Events Summaries	6
5	Brief Summary of Use Cases	7

List of Figures

1	Context diagram for Ultimate Calculator	5
2	Use Case diagram for Ultimate Calculator	7

Table 1: **Revision History**

Date	Version		Notes
February 2022	10,	1.0	Context, Use Cases and NFR Added
February 2022	10,	1.1	Work Partitioning and Project Issues Added
February 2022	10,	1.2	Project Drivers Section Added
February 2022	11,	1.3	Constraints, Naming Conventions, Functional Requirements Added
February 2022	11,	1.4	Symbolic Parameters, Cross Referencing
April 4, 2022		1.5	Rev 1 Updates

This document describes the requirements for the Ultimate Calculator. The template for the Software Requirements Specification (SRS) is a subset of the Volere template (Robertson and Robertson, 2012).

1 Project Drivers

1.1 The Purpose of the Project

Ultimate Calculator is a multipurpose calculator that computes different and useful things that a typical calculator can not compute. This includes things such as basic arithmetic, algebraic problems, and unit conversions. The user can select what they would like to use the calculator for and enter the parameters required by the system to complete the calculation. The purpose of the project will be to redesign the outdated GUI, improve the computational error handling and robustness of the algorithms currently in place, and add new features to improve the calculator functionality.

1.2 The Stakeholders

1.2.1 The Client

The clients for our project are Dr. Ashgar Bokhari, the instructor for SFWRENG 3XA3, Veerash Palanichamy and Oluwaseun Owojaiyo, the teaching assistants for lab section 1. The role of the clients is to instruct the team on what documents and deliverables need to be completed, provide help to students, and evaluate the project if it emulates the requirements found in the SRS.

1.2.2 The Customers

The customers of the project are students, educators, or any person who requires a multipurpose calculator. The main demographic are students, but the generality of the calculator will allow anyone to use it.

1.2.3 Other Stakeholders

The developers (Group 15) are stakeholders, as they are entirely involved in the creation of the project and directly influence the final outcome. Also,

any person who wants to improve upon the open source code of the Ultimate Calculator are also stakeholders, as they would have interest in the success of the end product.

1.3 Mandated Constraints

1.3.1 Solution Design Constraints

Description: The application must operate on any personal computer machine running Windows, Linux or Mac OSX

Rationale: ~~the~~ The user will access the application through a functioning personal computer via an operating system

Fit criterion: ~~the~~ The application will be made to run on any personal computer running Windows, Linux or Mac OSX

Description: The application must have the necessary input sections required to compute the mathematical solution

Rationale: The user should be able to input the necessary values to compute the solution

Fit criterion: The application will include the necessary input sections required to compute the solution

1.3.2 Partner or Collaborative Applications

The application does not have any partner or collaborative applications. The application relies on a functioning operating system to facilitate application launch and computation.

1.3.3 Off-the-Shelf Software

The application does not require any Off-the-Shelf software.

1.3.4 Anticipated Workplace Environment

The anticipated workplace environment of the application is anywhere. The application can be run from any personal computer. Personal computers can be both portable and stationary resulting in the application being used at any location without the need for an internet connection.

1.3.5 Schedule Constraints

The project will follow the predefined deadlines and due dates provided by the course instructor and the teaching assistants. The deliverables must be completed within their specific due dates and the project must be completed by April 12th, 2022.

1.3.6 Budget Constraints

There is no applicable budget for this project. The cost of development tools and hardware will be out of pocket costs to the developers.

1.3.7 Enterprise Constraints

The application will be free to download. Any user with a functioning personal computer with an operating system will be able to access the software without the need for an internet connection.

1.4 Naming Conventions and Terminology

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
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

Table 2: Naming Conventions and Terminology

1.5 Relevant Facts and Assumptions

1.5.1 Facts

- The original repository contains approximately 3000 lines of Python code.
- The application has an MIT license.

1.5.2 Assumptions

- Users have necessary equipment to run the application such as a computer, mouse, and keyboard.
- Users have basic knowledge of operating a computer.
- User have basic linguistic understanding of the English language.
- Users have basic knowledge of the mathematical concepts covered in the calculator section to input the values accordingly.
- Users have basic knowledge of calculating applications and their functionalities.
- Users are physically able to maneuver through the application via a mouse and keyboard.
- Users are visually able to maneuver through the application via mouse and keyboard.
- The application will be used by one user through their local machine.

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

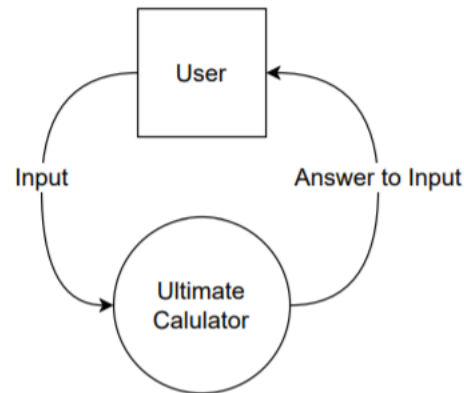


Figure 1: Context diagram for Ultimate Calculator

The Ultimate Calculator will be totally independent from any other external systems. There will only be a single user that interacts with the system.

2.1.2 Work Partitioning

Event Number	Event Name	Input	Output
1	Start The Application	Keyboard	Main Menu
2	Choose an Operation Type	Mouse Click	Operation Type Window
3	Choose Operation	Mouse Click	Operation Window
4	Enter a Value	Keyboard & Mouse Click	Value
5	Apply the Operation	Mouse Click	Final Answer
6	Clear Inputs	Mouse Click	Empty Input
7	Close Window	Mouse Click	Main Menu

Table 3: Work Partitioning Events

Event Number	Summary
1	The user will use their keyboard to prompt the application to start from their terminal.
2	From the main menu the user will have various options of operation types to click(i.e basic arithmetic, algebra, convertors, etc). Once a type is chosen the user clicks on the type they desire and the operation type window will appear.
3	From the main menu the user will have various options of operation types to click(i.e basic arithmetic, algebra, convertors, etc). Once a type is chosen the user clicks on the type they desire and the operation type window will appear.
4	From the operation window the user will click on the input box they desire and type in their inputs with their keyboard.
5	From the operation window the user can complete an operation and find a final answer. The user must have the inputs filled in and if they are they will click a calculate button. The user will be shown the answer they desire.
6	In the operation window the user can click the clear button found in the current window to clear all values they have previously input.
7	If a user chooses to close a window or close the application they can select the X button found in the top right corner.

Table 4: Work Partitioning Events Summaries

2.1.3 Individual Product Use Cases

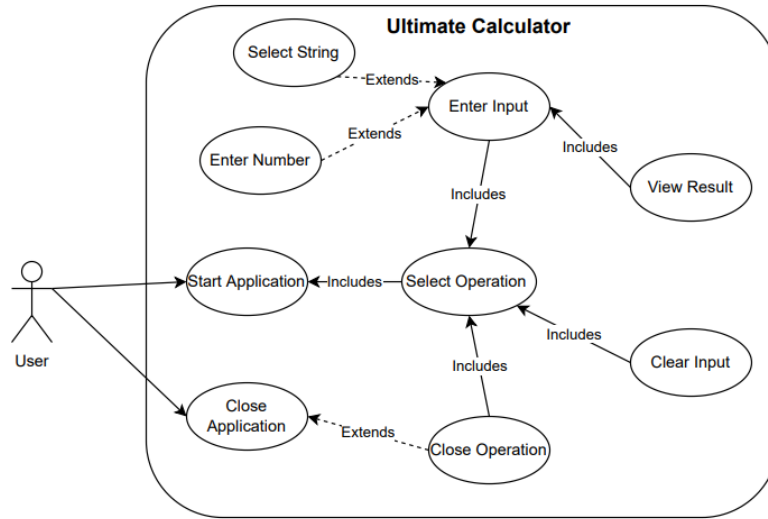


Figure 2: Use Case diagram for Ultimate Calculator

Actor	Actors Goal	Use Case
User	To open the application to begin computing.	UC1 - Start Application
User	To perform a specific computation.	UC2 - Select Operation
User	To enter the necessary parameters for the desired calculation.	UC3 - Enter Input
User	To see the answer of the computation.	UC4 - View Result
User	To remove previous inputs.	UC5 - Clear Input
User	To stop using a specific operation.	UC6 - Close Operation
User	To stop using the application.	UC7 - Close Application

Table 5: Brief Summary of Use Cases

Figure 2 illustrates several ways in which the user can interact with our system and the possible actions that they would want to perform. An overview of the main use cases can be seen in Table 5.

2.2 Functional Requirements

FR1 The system must display all operation types of the calculator upon the start of the application.

- **Fit Criterion:** The main menu has buttons that display all operation types.

FR2 The system must have at least *MIN_UNIQUE_OP* unique operation types.

- **Fit Criterion:** The system contains at least *MIN_UNIQUE_OP* unique operation types.

FR3 The system must have a minimum of *MIN_OP_SECTION* operation sections for every operation type.

- **Fit Criterion:** Operation types have at least *MIN_OP_SECTION* operation sections.

FR4 The system must start users at the main menu.

- **Fit Criterion:** The main menu is the first window displayed.

FR5 The system must initialize all input parameters for an operation to be empty upon user selection of that operation.

- **Fit Criterion:** Text boxes and other inputs are empty upon opening of an operation section.

FR6 The system must display a calculate button in every operation section.

- **Fit Criterion:** A calculate button is present in every operation section.

FR7 The system shall let the user compute multiple operations synchronously.

- **Fit Criterion:** Calculations in one operation section do not interfere with calculations in another.

FR8 The system must provide users with a way of entering input parameters.

- **Fit Criterion:** The system has text boxes or buttons for input parameters.
- FR9 The system must have a minimum of *MIN_INPUT* input for every operation section.
- **Fit Criterion:** The system has *MIN_INPUT* amount of text boxes or buttons as input.
- FR10 The system must display the input that the user has provided.
- **Fit Criterion:** The operation sections have a text box for user input.
- FR11 The system must tell the user if their inputs result in mathematical errors such as a division by 0 error.
- **Fit Criterion:** The system displays a error message when the user's input results in mathematical errors.
- ~~FR12 The system must prevent users from entering non-valid input types and provide a warning.~~
- FR12 The system notify users when non-valid input types are entered.
- **Fit Criterion:** The system incorporates error handling and return a message when the input is invalid.
- ~~FR13 The system must notify users if the necessary input fields are not populated~~
- FR13 The system must notify users if the necessary input fields are not populated at the time of calculation.
- **Fit Criterion:** The operation sections displays an invalid input message when the necessary inputs are not populated.
- FR14 The system must produce mathematically correct results based on well known arithmetic rules.
- **Fit Criterion:** The operation sections displays the answers in a textbox populated with mathematically correct values.

FR15 The system must display the results of a calculation to the user.

- **Fit Criterion:** The operation sections displays the answers in a textbox populated with mathematically correct values.

FR16 The system must allow the user to clear input.

- **Fit Criterion:** The operation sections display a clear input button that clears all user inputs when clicked.

FR17 The system must be able to clear all user input in the selected operation.

- **Fit Criterion:** The operation sections display a clear input button that clears all user inputs when clicked.

FR18 The system must allow the user to close an operation section.

- **Fit Criterion:** The operation sections display a close button on the top right corner of each window in addition to a separate operation button displayed in each section.

FR19 The system must allow the user to close the main menu at any time.

- **Fit Criterion:** The system closes the main menu window when the close button is clicked and confirmed.

FR20 The system must close all operation sections upon the closing of the main menu.

- **Fit Criterion:** The system closes all open windows when the close button of the main menu is clicked and confirmed.

FR21 The system must prompt the user with a confirmation window when they choose to close the main menu.

- **Fit Criterion:** The main menu window launches a confirmation prompt when the close button is clicked.

3 Non-functional Requirements

3.1 Look and Feel Requirements

Requirement NF1:

- **Description:** The main menu must resemble a standard calculator with various additions.
- **Rationale:** Keeps the appearance in line with the context of the application.
- **Fit Criterion:** The main menu has similar appearance traits to a real calculator such as an output screen and buttons.

Requirement NF2:

- **Description:** All sections of the application must be similar in appearance.
- **Rationale:** This will make the application look more cohesive and professional.
- **Fit Criterion:** Font type, sizing, and colour, along with background colours will all be consistent throughout the application.

3.2 Usability and Humanity Requirements

Requirement NF3:

- **Description:** All operation sections shall be easy to access.
- **Rationale:** The user wants to get to a specific operation section quickly and efficiently.
- **Fit Criterion:** The user should be able to access a specific operation section within *MAX_NAVIGATION_CLICKS* mouse clicks.

Requirement NF4:

- **Description:** Maneuvering through the menus shall be simple.
- **Rationale:** The user wants to be able to locate what they want intuitively.
- **Fit Criterion:** All navigation buttons will be specifically labeled or have descriptive images on them.

3.3 Performance Requirements

Requirement NF5:

- **Description:** The transitions from each individual menu should be a smooth and fluid process.
- **Rationale:** This will improve the experience of the users of the application.
- **Fit Criterion:** A menu should appear in a maximum of *MAX_RESPONSE_TIME* seconds after it has been prompted.

Requirement NF6:

- **Description:** The system must display the results of an operation quickly.
- **Rationale:** If a calculation takes too long the calculator will not be viable.
- **Fit Criterion:** The result must be displayed by the system in less than *MAX_RESPONSE_TIME* seconds from once the calculate button has been selected.

Requirement NF7:

- **Description:** The system shall display a maximum of *MAX_SIG_FIGS* significant digits for any calculation result.

- **Rationale:** The GUI only has a certain amount of space for displaying answers, so a number with too many significant digits may result in an overflow of numbers off the GUI.
- **Fit Criterion:** A maximum of *MAX_SIG_FIGS* significant digits is displayed by the system.

3.4 Operational and Environmental Requirements

Requirement NF8:

- **Description:** The system must work without the use of an internet connection.
- **Rationale:** To allow the user to use the application anytime and anywhere.
- **Fit Criterion:** The application works without access to the internet.

3.5 Maintainability and Support Requirements

Requirement NF9:

- **Description:** The product must be made open source where anyone can submit bugs and issues.
- **Rationale:** This will allow for any changes or issues to be easily tracked.
- **Fit Criterion:** The product is open source on a website such as GitLab or GitHub.

Requirement NF10:

- **Description:** The system must allow new operation types to be easily added.
- **Rationale:** If updates are to be made in the future, this will be necessary.
- **Fit Criterion:** The code is modular and exhibits low coupling and high cohesion.

3.6 Security Requirements

N/A

3.7 Cultural Requirements

N/A

3.8 Legal Requirements

N/A

3.9 Health and Safety Requirements

N/A

4 Project Issues

4.1 Open Issues

There were not any issues in the implementation of the Ultimate Calculator. The downloadable executable file launches the console based calculator but the GUI requires additional setup to work.

4.2 Off-the-Shelf Solutions

[Symbolab](#) is the best known answer engine found on the internet, but it is not an open source project like our Ultimate Calculator. [Symbolab](#) contains several advantages that allow the user to do various calculations most open source calculators can not do. Yet, [Symbolab](#) is less convenient as it can not function without the use of the internet. Another off-the-shelf solution would be a handheld calculator. The handheld calculator can function without the use of the internet but it can not do the advanced calculations that the Ultimate Calculator can.

4.3 New Problems

Additional operations such as ~~matrix manipulation~~ BMI calculations, binary calculations and a GPA calculation will be added. A new and improved GUI will also be a part of the project.

4.4 Tasks

1. Design templates of the new GUI look for all sections of the application.
2. Modularize the source code further.
3. Make original source code more robust when it comes to erroneous inputs.
4. Implement the new GUI designs within our application.
5. Implement new functionality and operations for the calculator.
6. Implement GUI designs for the new operations within our application.
7. Complete all course deliverables on time.

These tasks can be seen in greater detail in the project Gantt Chart: https://gitlab.cas.mcmaster.ca/petronim/ultimate_calculator_l01_group15/-/tree/main/UltimateCalculator/ProjectSchedule/3XA3ProjectPlan.pdf.

4.5 Migration to the New Product

The new Ultimate Calculator will use elements of the previous implementation such as backend calculations, but a new GUI will be used.

4.6 Risks

There is little to no risk with this project as the mathematics underlying the calculator are already well established and well documented. The risks of the program include imprecise answers to calculations and unexpected errors such as divide by zero. These risks will be mitigated by thorough testing.

4.7 Costs

There is no monetary cost associated with the development of the application. The development of the application is estimated to consume a time cost and requires prerequisite software and hardware that the developers must provide for themselves.

4.8 User Documentation and Training

4.8.1 User Documentation

There will be a README file available that guides the user through the installation process.

4.8.2 Training

No training is required to use the Ultimate Calculator application. The selections are intuitive and a basic understanding of how to input values is required.

4.9 Waiting Room

If time is found for additional implementations these ideas would be considered.

1. Integral Calculator
2. Derivative Calculator
3. Graphing Calculator
4. [Mobile implementation](#)

4.10 Ideas for Solutions

N/A

References

James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.

5 Appendix

This section has been added to the Volere template.

5.1 Symbolic Parameters

MAX_RESPONSE_TIME = 2
MAX_NAVIGATION_CLICKS = 2
MAX_SIG_FIGS = 64
MIN_UNIQUE_OP = 5
MIN_OP_SECTION = 1
MIN_INPUT = 1