

Table 1: Revision History

<b>Date</b>	<b>Developer(s)</b>	<b>Change</b>
February 3, 2022	Mathew Petronilho	Added Introduction, Meeting Plan and Project Schedule
February 3, 2022	Jarod Rankin	Added Git Workflow plan and Technology
February 3, 2022	Logan Brown	Added Team Communication Plan and Proof of Concept Demonstration Plan
February 3, 2022	Syed Bokhari	Added Team Member Roles and Coding Style
February 4, 2022	Mathew Petronilho	Punctuation and Grammar Update
April 4, 2022	All Members	Rev 1 updates

# SE 3XA3: Development Plan Ultimate Calculator

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

The following document will contain the development plan for our Ultimate Calculator project, including team organization and implementation of the project.

## 1 Team Meeting Plan

### 1.1 Meeting Schedule

Where	When	Frequency
Microsoft Teams	9:30 am to 11:30 am	Every Tuesday and Thursday
Discord	8 pm to 10 pm	Every Monday

Table 2: Team Meeting Schedule

### 1.2 Meeting Roles

1. Note Taker: This person's role is to record any key decisions that have been decided on, any problems that have been encountered, and keep track of results of the meetings. They must ensure that all these results are shared with the other members through well maintained documents.
2. Time Keeper: This person's role is to keep track of meeting dates and durations and record them into a shared document. They must also ensure the meeting schedules are followed and that time limits for specific topics are respected.
3. Group Atmosphere Monitor: This person's role is to watch the tone of the conversation and ensure the topic of the meetings stays project related.

They will help to settle conflicts and speak up when things get out of hand.

4. Visionary: This person's role is to ensure what is being done is helping the project move along and contributes to overall productivity. They will guide the group through the discussion and make sure that everyone is participating.
5. Participants: This person's role is to punctually attend each meeting and to contribute to the workflow of the project. They must be respectful and communicate effectively and concisely.

### 1.3 Agenda Rules

The agenda will contain topics to discuss for a particular meeting, along with the allotted times for each topic. If any activities or resources are required for a part of the meeting, they will be recorded in the agenda. Attendance will also be recorded in the agenda at the start of each meeting.

## 2 Team Communication Plan

All meetings for this project will take place over Microsoft Teams or Discord as discussed in the meeting plan. Outside of meetings, all communication will take place over a private chat on Discord. Team members are expected to check this chat at least two times a day.

Name	Microsoft Teams / Gitlab	Discord
Syed Bokhari	bokhars@mcmaster.ca	Fyke
Mathew Petronilho	petronim@mcmaster.ca	Matt_Petro
Jarod Rankin	rankij5@mcmaster.ca	jarodr11
Logan Brown	brownl33@mcmaster.ca	1brownlog

Table 3: Member Contact Information

### 3 Team Member Roles

The following table highlights the assigned roles for each member of the team:

Name	Role(s)
Syed Bokhari	Participant Note Taker Developer Tester Git Expert
Mathew Petronilho	Participant Visionary Developer Tester Python Expert
Jarod Rankin	Participant Group Atmosphere Monitor Developer Tester Documentation Expert
Logan Brown	Participant Time Keeper Developer Tester Design Expert

Table 4: Member Roles

### 4 Git Workflow Plan

Our project will follow a feature-branch model where each new feature is pushed to their respective branches. These branches will be merged with the main branch only when the feature is complete and functional. The branches will be updated with the main branch daily to prevent future merge errors. Every commit made to the ~~repo~~ repository should have a descriptive message to identify what each commit has changed. All final commits for a project milestone will be tagged in accordance with the tag provided in class.

### 5 Proof of Concept Demonstration Plan

#### 5.1 Significant Risks

##### 5.1.1 Implementation

The implementation of the GUI will be the most difficult part as we don't have much experience creating our own. The backend side of the project should not

be as difficult since we already have the domain knowledge for the calculations.

### **5.1.2 Testing**

Testing the correctness of the calculations is not a concern since those always have a single acceptable answer. Testing of the GUI however is more difficult as there are many unpredictable ways a user can interact with a GUI which we may not catch.

### **5.1.3 Library Installation**

The GUI library PyQt5 is well known and should be straightforward to install.

### **5.1.4 Portability**

Portability is not a concern since python and the associated libraries being used are cross platform.

## **5.2 Overcoming Risks**

### **5.2.1 Implementation**

To overcome the risk of difficulty in implementing a GUI, we will use a standard library PyQt5 which is known for being simple to use and understand. This library was used in the previous implementation which we can use to learn how the library is used and how we can improve it. We will also conduct our own research on how to use PyQt5 through tutorials and online resources.

### **5.2.2 Testing**

Testing of the GUI can be achieved mainly through manual testing. To ensure a large variety of cases are tested, all team members will conduct their own tests individually. In addition to this, we will conduct both functional testing (calculations, I/O, dividing by zero, etc.) and non-functional testing (responsiveness, ~~resizing window~~, ease of navigation, [appearance](#) etc.)

### **5.2.3 Library Installation**

All team members will have to install the necessary libraries for testing purposes, so if any issues arise team members can help each other out.

### **5.2.4 Portability**

Team members will use the same, up to date versions of python and necessary libraries and ensure that they are all cross platform.

## 6 Technology

### 6.1 Programming Language

Our project will be written in a current version of Python. Python was chosen as all group members have familiarity with the language and Python's various libraries can help us in implementing a better design.

### 6.2 IDE

The IDE most group members will be using is Visual Studio Code, as it is easy to use and it allows us to commit and push our code directly from the application. Group members may also use a combination of various other IDEs and text editors such as PyCharm and Sublime Text.

### 6.3 Documentation

For documentation purposes the group will be using a combination of Google Docs and Overleaf to produce L<sup>A</sup>T<sub>E</sub>X documents. These documents can be found in the project repository on GitLab. Any added features to the source code will also be documented using doxygen.

### 6.4 Testing

To test our project we will use ~~pytest~~ python's `unittest` library due to the language of choice and the familiarity for each group member. We can also test our GUI components through various manual testing practices.

## 7 Coding Style

The code will follow the style specified by Python.org. This can be found at <https://www.python.org/dev/peps/pep-0008/>. The guidelines will be used to create an easily readable environment with the addition of code simplicity. The variables `and methods` will be named using camel case, while the classes ~~and methods~~ will be named using pascal case. Comments will be used to document the code as it's written and will be updated in accordance with code changes. The comments should be easily understandable and limited to a length of ~~72~~ 100 characters. Complete sentences will be used starting with a capital letter.

## 8 Project Schedule

The project schedule takes the form of a Gantt Chart and can be viewed here: [https://gitlab.cas.mcmaster.ca/petronim/ultimate\\_calculator\\_101\\_group15/-/tree/main/UltimateCalculator/ProjectSchedule/3XA3ProjectPlan.pdf](https://gitlab.cas.mcmaster.ca/petronim/ultimate_calculator_101_group15/-/tree/main/UltimateCalculator/ProjectSchedule/3XA3ProjectPlan.pdf). This schedule will be updated regularly through out the project duration as deliverables become more clear.

## 9 Project Review

For our project, we as group 15, tackled the task of improving the open source calculator application, the Ultimate Calculator. As a team we have come to the conclusion that the project was an overall success. When first met with the source project the team quickly recognized there were many faults to the original source code that could be improved upon. To begin, the first area of improvement we wanted to tackle was updating the GUI, making it look more modern and cohesive. Secondly, we wanted to make the program more robust, the original source code had no testing done which also meant there was no error handling done, the program was not robust at all. And finally our third main goal was to increase the usability of the application, as the original source project was hard to navigate.

The team tackled these problems efficiently throughout the duration of the project. Firstly, the team used the python based GUI toolkit PyQt5 to improve the GUI of the application. The group made sure each window of the application was updated to look more modern, this was done by changing the colours to resemble the McMaster school colours, and that each window was consistent and cohesive by using the same GUI asset styles, fonts, text sizes and colours. Secondly, the team made sure to create unit tests for each module that contained calculator functionality and created error handling to make the application more robust. Each operation window now handles invalid inputs and division by zero errors. Finally, the project team increased the usability of the application by placing the main calculator on the main window to resemble a real calculator, as apposed to being buried somewhere in the application as it is in the source project. Additionally, each operation interface is given a clear button to clear input fields so the user does not have to manually clear each field one by one. Also, upon close of an operation window the input fields are also closed as opposed to the source project where when the window is closed, the input fields hold the value of whatever was last in there.

Additionally the team added four new operation types to the application, a GPA calculator, a health calculator, which includes a body mass index calculation and a body fat percentage calculator, a binary calculator, and a geometry calculator. These additional features helped improve the overall usability of the project, giving the user a variety of new calculations they can do. One change the team would like to do if we were to continue with the project is add more complex functions, such as matrix calculations. Additionally the team would like to expand the main calculator by adding trigonometric functions such as sine, cosine and tangent.

The final project that was presented to the class fulfilled all the requirements that are stated in Software Requirements Specification. Although when revising the document, we found that some requirements we had set were unrealistic based on how we had completed the project so far. These requirements were

updated and consistent with the final projects source code.

The team dynamic grew and became stronger as the project progressed. The team used Discord to plan and hold meetings where the team would work together and distribute work that was needed to be completed for the deliverable that was due next. Even though the team had to work on this project through the unpredictable times of the Covid-19 pandemic the team managed to be efficient and work together coherently.

In conclusion, as a collective, we Group 15 believe the project was an overall success. Throughout the semester the group has learned a lot about the software development process, the documentation process and refined our collaboration skills. The group is proud to have their names on the final product they have created and have come out of this experience better software engineers than when we began the course.