Team F - ETERNITY Calculator

Castonguay, Justin — Fakhr, Daniel — Hernandez, Jaime Andres
Thibault-Shea, Daniel — Yaghma, Ashkhan

August 5, 2019

Contents

1 Team organization and collaboration patterns

1.1 Lead up to Iteration 2

1.2 Meeting Schedule

We agreed to meet at least once a week for 2 hours. We are also in constant contact on Discord and we all keep apprised of recent pushes to the Git repository to stay up to date with the advancement of the code base and the accompanying documents. Detailed meeting minutes are posted to the Git repository after every meeting by the designated secretary for that meeting. We are also using Google Docs to collaboratively work on draft documentation, which can then be used to create our final documentation with the correct formatting.

1.3 Our Strategy for Iteration 2

In order to develop a working software calculator prototype in the coming weeks, our team has come up with a development strategy to be prepared for challenges that we may face. This strategy must take into account a plan for writing requirements for features implemented, the technologies selected to develop the calculator, ideas of algorithms for numerical computation of selected functions, and tasks to be allocated to each team member based each of our strengths. With a well advised plan of action, we are much more likely to collaborate efficiently as a team and ultimately meet the deadline of the project.

1.4 Task assignment

During the second team meeting we went over the requirements for deliverable 1 and made a list of all of the things that needed to get done. We quickly saw that this was a significant amount of work and that there were some dependencies between the tasks (ex. do interviews before use cases). We divided the tasks into 3 categories: 1) Information gathering, 2) Consolidation/Report, 3) Coding and prototyping. A cursory evaluation of the workload vs. the deadline showed that we could not possibly deliver all of these tasks on time. Rather than cut out the early-stage prototyping entirely, we decided on making a priority matrix of our tasks with the simple rule that higher priority tasks should be completed before an individual started work on the lower priority ones.

We reasoned that the top priority items were those on the critical path of deliverable 1. We all had a good idea of what a calculator should be but we also knew better than to make a product for the developers. Consequently, we absolutely needed the information from the interviews as soon as possible so that we could align our efforts with what the calculator's potential users wanted. This step was so critical and urgent that we put 4 people on it with highest priority.

We then consolidated and paraphrased the interview data. We were surprised at just how much information we got from 5 interviewees. Firstly, we extracted the key features each user wanted. Secondly, we distilled down each interview into the key themes that were important for that interviewee. Thirdly, we were able to classify our interviewees based on their expected usage of the calculator (basic use, mathematical use). Lastly, we looked for commonality across the different wants of the interviewees and obtained what we think is a much better approximation of what the market wants from a calculator.

One team member was tasked with coming up with an outline of our testing strategy. We decided that since the users all seemed to value accuracy of the mathematical function on the calculator then a test-driven approach to development would be appropriated.

Another person was tasked with researching the algorithms needed for the implementation of the non-trivial calculator functions. Some functions had multiple algorithms that varied in complexity. For this first iteration, it was decided that simplicity should be the key criteria in the choice of algorithm since the deadline was so tight. We all agreed that we could reevaluate this strategy for the next iteration.

The very last priority was the implementation of the mathematical functions. Some team members were eager to start coding but we decided that it would be a much better idea to focus on the requirements gathering at this stage of development. Consequently, only a very rough implementation

of some of these functions appear as code.

Table 1: Task Priority Matrix

Name	**Priority 1**	**Priority 2**	**Priority 3**
Ashkhan	Interviews/personas	Testing strategy (TDD)	a^x, 10^x
Daniel F.	Interviews/personas	Algorithm research	x^y, $\sqrt[n]{x}$
Daniel T.	Coordinate activities	Use case diagram/desc	e^x
Jaime	Interviews/personas	Interview summary	$cos(x)$, $sin(x)$
Justin	Interviews	Strategy	$ln(x)$

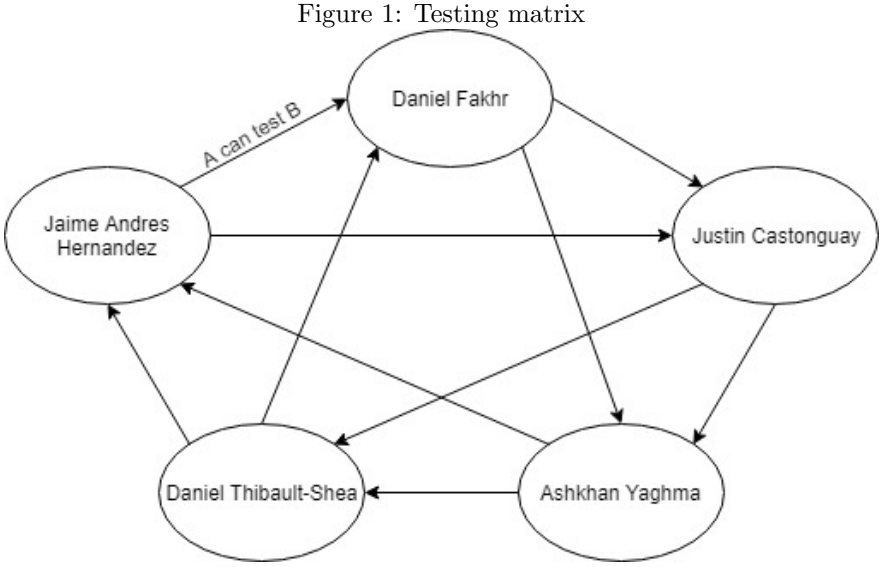
1.5 Testing strategy

For testing purposes, we decided to use Junit to write all of our unit tests. These tests would verify the functionality of our different mathematical functions. We would need to think of and write up as many test cases as possible for different scenarios (negative numbers, large and small numbers, invalid input, etc) in order to cover all potential issues.

Test coverage is incredibly important in the early stages of a software project. Early and thorough testing reduces the risk of having an insurmountable amount of bugs later on. Taking this into consideration, we decided to employ a Test Driven Development strategy in which we would write unit tests before the functions were complete and imposed a rule that all valid tests must pass before any team member makes a push to our master branch. This would ensure that whenever anyone pulled from the master branch, they would not have to waste time fixing someone's bugs to work on their own task.

We all agree that testing each other's code is of paramount importance to the project. To not do so would lead to colossal wastes of time in tracking down myriad bugs. We made the following testing matrix which will ensure that there are no conflicts of interest in the testing. The arrow indicates who can test who - no arrow means you cannot test that person's code.

Figure 1: Testing matrix



The source code peer review was organized such a way that no pair of people review each others code.

The code review was done in the similar order :

Jaime Andres Hernandez → Daniel Fakhr → Justin Castonguay → Ashkan Yaghma → Daniel Thibult-Shea → Jaime andres Hernandez.

The following table summarize some of the code reviews that were undertaken and the associated comments provided to the reviewees.

Reviewer	Reviewee	Comments
Jaime Andres Hernandez	Daniel Fakhr	-Try to follow Proper formatting (some of your code is not properly indented) -Try to comment more, some functions are missing comments. -Use JavaDoc comments so we can auto-generate docs
Daniel Fakhr	Justing Castonguay	-Use more comments. Some of your comments are uncommented -Test your functions. Some of them don't work as they should -Use "throws Exception" in the method declaration when not catching the Exception within the method.
Justing Castonguay	Ashkan Yaghma	-Format your code better -Use better variable names
Ashkan Yaghma	Daniel Thibault-Shea	-Don't leave commented code. -Comment your functions.
Daniel Thibault-Shea	Jaime Andres Hernandez	-Write JavaDoc comments (including comments for constants) -Elaborate more in your comments -Try to make your code more efficient

2 Requirements Gathering

In order to create software requirements, our team got together to brainstorm ideas for what features the calculator would have and how to implement them. We made sure our features were realistic, keeping in mind the time constraint of the project and the development experience of the team. Once we had a base for how we thought our calculator could be developed, we decided to conduct interviews on potential users to find out what features everyday users of calculators actually valued and if our initial ideas complemented these.

From the answers from our interviewees, we created user personas that had concrete problems and tasks that needed to be completed and that our software would solve. The user personas along with our initial discussions would then inform what different use cases might be for our calculator. The use cases were put into a standard use case diagram so that the high-level design of the calculator could be understood at a glance. Each use case was then expanded in a summary use case description.

2.1 Common interviewee points summarized

Sarah
HR Student
- inputting full eqn like she memorized
- physical calculator
- simple calculator
- prefers physical calculator but uses others
- functions in the book
- downloadable functions or packages
- cares about precision and the right answer
- doesn't care about aesthetics
- hot keys for common functions

Victoria Benlala
Entrepreneur, Spa Owner
- Button to calculate the taxes (simple programmable functions)
- phyiscal calculatior first doesn't mind others
- doesn't use complicated functions
- simplicity and soft buttons. Would like a more portable version.
- mapping numbers to number keys on computer. Being able to have hot keys or set them up himself with the functions he or she is given

Kevin
Engineering student
- Would like to be able to access functions easily for engineering
- Comfortable with both software and hardware but prefers hardware

- He wants shortcuts
- Wants basic functions also
- Wants to use computer keyboard and not mouse pointer
- Portable and key mappable
- Use symbols that are already commonly found on calculator on the cpu keyboard also
- Include a shortcut quit key
- Hot keys (like S for sin, T for Tan, etc.)
- Recommends skins for calculator
- Would like downloadable packages for functions to customize calculator

Tarek
Electrical engineering student
- accuracy, speed, and comfort
- basic essential functions
- he would like it to be able to plot graphs
- he would like to transfer his work from calculator to mobile
- prefers physical but he uses other for quick calculations
- wants calculator easy to hold
- would like it to be cheap even if it's customizable

Arash
Avionics Engineering Student
- specific buttons for each function
- prefers an app
- He would assign each function to a specific button

2.2 Common Ideas

- Simplicity
- Physical calculators → GUI could look like physical calculator
- Hot keys
- Simple functions (plus, minus, etc.)
- Portability
- Customizable (physical and software wise) download functions

2.3 Interview Data Summary

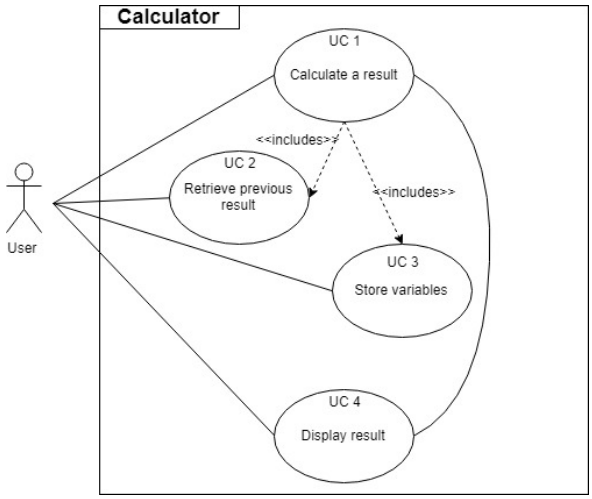
Looking through all the interviews we were able to pick up on some important points that we chose to consider when creating our use case diagrams and to move forth with our project. We interviewed a Human Resource, Mechanical Engineering, Electrical Engineering, and Avionics Engineering student. We also interviewed an Entrepreneur/Spa Owner to gather our data. Each individual had very different needs specifying what kinds of functions they would like to see on their ideal calculator. For example, Engineers wanted integration functions, while an entrepreneur wanted percentages or tax calculating functions.

What they all had in common though was the want for simple operations (like addition, subtraction, etc.). They all also wanted simplicity in terms of the calculator's look, how easy it would be to access the functions they wanted to use, understand what they are, and it's portability. The majority also wanted a reliable calculator in terms of precision and accuracy. They all preferred physical calculators over software calculators (like those you would find on a computer as an extra tool application). They all liked the idea of mapping keys on a computer keyboard to their desired functions to make the calculating experience more personalized and simple to them. They all had different ways they wanted to customize their calculator, which included personalizing it physically and software-wise. The important point though was that customizability was what they valued commonly amongst each other.

Based on the research we made from the stakeholders we interviewed, the calculator will need to be **simple**, **customizable**, and **reliable**.

2.4 Use cases

With the initial requirements gathering step mostly completed, we were now able to



3

4 Prototyping

4.1 Source code overview

This iteration was mostly marked by requirements gathering mainly through the interview process. A limited amount of research was done into evaluating different algorithms for implementing of some functions.

While that was ongoing, we thought that we needed something concrete to present at this stage. While nothing look even remotely like a calculator at this stage, several function were implemented satisfactorily.

For example, $ln(x)$, $sin(x)$, e^x, and $\sqrt[n]{x}$ were implemented with some measure of success. Choices were made to used the simplest algorithm at this stage and to only test over a certain range of inputs. The full implementation will be done for the next iteration.

Following the interviews with the engineering students, we decided on implementing some extra functionality such as $x!$, binomial coefficient, etc. to better satisfy this class of users's needs. We were able to justify their inclusion at this early stage in part due to their relatively simple implementations.

There are still an immense amount of implementation details that the team needs to agree on at this stage. The interview process gave us so many ideas and potential requirements/features that we have not had enough time to debate on. We reserve the resolution of these issues for the next iteration. At this stage, it is exciting to see how the product is taking shape. With every meeting we feel we are getting tangibly closer to the final version of the product. We know that we will have some hard decisions to make very soon and that there may be some slight head butting. Regardless, we are all keen on compromising and getting a good product out the door.

4.2 Algorithms evaluation

Based on our experience and with the information gleaned from the interview, we decided to implement at least the following transcendental functions: x^y, e^x, $sin(x)$ and $cos(x)$, $tan(x)$, $ln(x)$ and $sqrt(x)$. Currently, optional functions are $arcsin(x)$, $arccos(x)$, $arctan(x)$. In order to develop solutions for our calculator without help from Java's math library, we would have to do research on numerical methods to compute these functions. We would have to find the sites with mathematical references such as Wolfram-Alpha, Wikipedia and YouTube in order to find solutions that would be feasible for us.

A few of these functions ($sin(x)$, $cos(x)$, $tan(x)$, e^x, and the $arc*(x)$ functions could be approximated relatively easily by Taylor polynomials. Series lend themselves naturally to iterative methods (ie. simple 'for' loop) so that is what these algorithms are.

For example, here is the algorithm for e^x:

$result \leftarrow 0$
$precision \leftarrow p$ (arbitraty precision integer)
$iterator \leftarrow 0$

while $iterator < precision$ **do**
  $result \leftarrow result + \frac{x^i}{i!}$
  $i \leftarrow i + 1$

return $result$

This algorithm does not pass test for a wide range of values of e^x. Currently, we are debugging this. This is an extremely simple algorithm compared to the other but they mostly take this form. The complexity of this particular algorithm is constant, however, it calls the x^i function which we have implemented hamfistedly as x multiplied by itself i times. We intend to optimize the power function in later iteration to take advantage of the many optimizations possible (ie. bitshifting, etc.).

The algorithm for $ln(x)$ is:

if $result \leq 0$
  $return$ **error**

$precision \leftarrow p$ (arbitraty precision integer)
$iterator \leftarrow 0$

while $iterator < precision$ **do**
  $result \leftarrow result + \frac{1}{(2i+1)} * \frac{(x-1)}{(x+1)}^{(2i+1)}$
  $i \leftarrow i + 1$

return $(2 * result)$

The $ln(x)$ algorithm is of a similar vein as e^x and of a similar complexity. Tests have been successful up to about x = 870. We are still debugging this. We think it has something to do with our algorithms being centered around 0 and the solution becoming less and less precise as we increase or decrease x away from 0. We will definitely have this sorted out for iteration 2.

4.3 Exclusions

Between the requirements gathering and the work we have already done on the prototype calculator we feel we are in a much better position to evaluate what features and functionality we can currently commit to (see task matrix) and which we must exclude. The following features were discussed and rejected for reasons that are documented in brief in the following table.

Feature	**Reason**
Distributed on web	No one has web skills. Everyone knows Java.
User customizations	We are not able to determine the workload that this represents since we are not far in the development of the calculator - TBD.
Mobile	Team does not have the skills to do that.
Hyperbolic functions	We don't even know what they are. The engineering interviewees never really used them except for that one class. We don't think they are worth the effort.
Distributed on web	No one has web skills. Everyone knows Java.
Animations	No one has any skills with that. This will be the first GUI for some of us.

The factor that most limits our willingness to undertake new functionality and features is the lack of skills found within the team. We all have the best intentions and want to provide a fully featured calculator but we had to draw the line somewhere. Currently, we are managing this by concentrating on the deliverable and focusing on the key features that the interviewees brought up.

5 Appendices

5.1 Appendix A - Interview Questionnaire

Suggested Interview Questions ETERNITY Calculator - Team F

Name: _____

Occupation: _____

Suggested Interview Questions

1. What do you use a calculator for?
2. What would you like your calculator to do? Or what is the ideal calculator for you?
3. What kinds of calculators have you used? (physical, apps, online, etc.). Hardware or Software?
4. What functions do you use most often? Which ones do you use the least?
5. What features did you like most about your calculator. What do you not like about your calculator?
6. Is the aesthetics of your calculator important? What matters most (shape, color, personalized themes, etc.)?
7. If you're using a keyboard, how would you map the keys to functions, numbers, etc. ergonomics?

5.2 Appendix B - Interview transcripts

Interview Q&A - 1st year HR student

What do you use a calculator for?

- Almost exclusively for her accounting and finance classes
- Most of her usage of the calculator is for simple lightweight math used in accounting and finance.
- She does not use it extensively, mostly for exams and homework.

What would you like your calculator to do? Or what is the ideal calculator for you?

- She likes her calculator to be simple
- Prefers a calculator that has its function symbols identical to the ones in the books
- She would like her calculator to display the answers in a human readable form (5x7 Matrix numeric representation), and not the digital form (7-segment numeric representation)
- In the future she would like to see a calculator network system, similar to that of the iclicker, where the professor would give you a password, which after inputting it into the calculator, downloads a custom function from the professor's base station, or unlocks/locks some of the pre programmed functions in the calculator.

What kinds of calculators have you used? (physical, apps, online, etc.). Hardware or Software?

- Uses both a physical calculator as well as a calculator app on her phone.
- Mainly uses the calculator application because it's always available.
- Prefers the physical calculator since she enjoys the tactile feel of the buttons and because phones are not allowed during exam time.

What functions do you use most often? Which ones do you use the least?

- Most of the time she uses functions such as: addition, subtraction, delete (backspace) , 10^x, Ans function for retrieving the previous answer , exponential and square root.
- Rarely, if ever, uses logarithms or any of the trigonometric functions.

What features did you like most about your calculator. What do you not like about your calculator?

- She feels indifferent about what she likes in her calculator. All she cares about is that the calculator gives the right answer.
- The only thing she does not like about her calculator is that it displays numbers in a digital format (7 segment numeric representation)

Are the aesthetics of your calculator important? What matters most (shape, color, personalized themes, etc.)?

- Most of the physical aesthetics are of little importance to her. What matters the most is that the calculator gives an answer.
- She claims that the physical aesthetics would be merely a perk and would not pay extra for them.

If you're using a keyboard, how would you map the keys to functions, numbers, etc. ergonomics?

- She would like her calculator to have some shortcuts to common functions such as percentage
- Prefers the shortcuts to have their own dedicated buttons on the calculator instead of having to press multiple buttons at once.

Thoughts on making a calculator that has customizable capabilities?

- She would like to have the ability to program functions into the calculator, but only if academic establishments would allow it.

Interview Q&A - 3rd year mechanical engineering student

What do you use a calculator for?

- University studies
- Mostly finds himself using it for mathematical purposes.

What would you like your calculator to do? Or what is the ideal calculator for you?

- His ideal calculator would not be missing essential functions like plus, minus, multiplication, exponents, and the basics. Other common functions he mentioned from University, include logs, derivatives, e, integrals, roots, square roots, and more. He believes it's a must to have the ANS (answer button), and calculation history included too.
- The calculator should be lightweight and portable.
- In the future he would like to see more calculators that provide more support for polar coordinates. Lastly he would like to have access to shorter readable manuals or video content to quickly go over all of the calculator's features and how to implement them.
- Believes It would be great to have the answer button, calculation history, polar capability and a shorter readable manual!

What kinds of calculators have you used? (physical, apps, online, etc.). Hardware or Software?

- Both softwares and physical.
- Prefers physical calculator more than software-based because he has become used to it since his High School years.
- He doesn't mind using software calculators as long as it has useful shortcuts. He defined a simple software calculator as something that can open and run quickly on a computer or any other device.
- Kevin also mentioned he uses software calculators mostly for basic problems but not derivations and more advanced operations because it becomes too tedious to work with. He would much rather use the physical one.

What functions do you use most often? Which ones do you use the least?

- Addition, subtraction, multiplication, division, exponents, roots, converting to fractions, sin, cos, tan, exponents, logs, and mod.

What features did you like most about your calculator. What do you not like about your calculator?

- He liked Hexadecimal conversion, octa, binary conversions, derivations, and integration functions because they're relevant to his engineering courses. .
- Kevin doesn't like using the mouse pointer on his computer to input the values and functions on his software-based calculator. He would much rather use the computer keyboard. He mentioned the clicking option should be removed completely to encourage others to use the keyboard.

Are the aesthetics of your calculator important? What matters most (shape, color, personalized themes, etc.)?

- Easy to fit in your palm, portable, not too colourful (greyish, black). It should also be key mappable if it is a software-based calculator.

If you're using a keyboard, how would you map the keys to functions, numbers, etc. ergonomics?

- He would use the numbered keypad for inputting numbers
- Any symbols that are already commonly found (+,-,;etc.) on both computers and physical calculators should be included.
- Assign important features and functions to large buttons, like the space bar or return key.
- Include a shortcut to quit
- Use the first letter of functions to input them into calculator. Ex: C for cos, S for sin, T for Tan, etc.

Thoughts on making a calculator that has customizable capabilities?

- Doesn't think it is necessary. He usually uses google to help solve complex problems and inputs the simple calculations on the calculator to double check his work and find the final answer.
- He believes it would be great to include physical skins to personalize the calculator but it is not a must.
- In the future he would like to make it customizable by being able to download packages for calculator functions that can easily be added or removed to the device.

Interview Q&A - 3rd year electrical engineering student

What do you use a calculator for?

- Any mathematics courses in his degree
- Almost all Engineering courses.
- Counting money at his job

What would you like your calculator to do? Or what is the ideal calculator for you?

- It should have the basic, essential functions such as square root, exponential, logarithms, and trigonometric functions.
- His ideal calculator would also have the ability to find variable unknowns (system of equations)
- He believes that a calculator that can plot and display graphs would be very beneficial
- Another feature he would like to see in calculators is the ability to calculate indefinite integrals.
- One of the main features he really wants to see is the ability to save/transfer his work (Graphs, functions, answers) from his calculator to his mobile phone.

What kinds of calculators have you used? (physical, apps, online, etc.). Hardware or Software?

- He has used all kinds of calculators, physical ones, apps, software...
- Prefers a physical calculator since he's used to its layout and buttons
- For quick calculations, he uses any calculator closest to him.

What functions do you use most often? Which ones do you use the least?

- Frequently uses functions such as trigonometric functions, exponential, logarithms, and root
- Hardly ever uses the modulus or absolute value functions.

What features did you like most about your calculator. What do you not like about your calculator?

- He dislikes that his calculator does not support finding unknown variables in equations.
- Claims that the few functions his calculator has for Radians and polar equations have been very helpful.

Are the aesthetics of your calculator important? What matters most (shape, color, personalized themes, etc.)?

- Aesthetics of his calculator are important to him.
- He prefers his calculator to be comfortable to hold
- Personalized themes on his calculator have little importance to him
- Prefers to buy a nice looking calculator that he likes and sticking with it instead of buying a customizable one.

If you're using a keyboard, how would you map the keys to functions, numbers, etc. ergonomics?

- He has no preference for key mapping, instead he would like his calculator to be set up in a way such that when he types the first few letters of the function name, it would show him function suggestions of which he can chose one.

Thoughts on making a calculator that has customizable capabilities?

- Would only buy it if it comes at no extra cost, otherwise he would prefer to buy a cheaper calculator with the pre-programmed functions that he knows he needs.

Interview Q&A - 2nd year avionics student

What do you use a calculator for?

- He uses it to solve his math and physics problems most of the times.
- He also uses it to keep track of his finances and plan his spending accordingly.

What would you like your calculator to do? Or what is the ideal calculator for you?

- In addition to basic functions he would like it to be able to calculate more complex functions such as e^x, log, roots, derivatives, integrals and so on.
- His ideal calculator is the one that has specific button for each function while it is portable.
- Maybe a folding calculator.

What kinds of calculators have you used? (physical, apps, online, etc.). Hardware or Software?

- He has used both hardware and software.
- He prefers using his engineering calculator app on his cellphone but since he is not allowed to use it in the exams he has to use his regular calculator most of the times.

What functions do you use most often? Which ones do you use the least?

- These days in addition to basic functions like multiplication and division, he mostly uses functions such as power, exp, root, log, trigonometric, derivative and integral.
- The factorial function might be one of those that he used the least recently.

What features did you like most about your calculator. What do you not like about your calculator?

- Since it is a pretty simple calculator, it is simple to use for the fundamental functions.
- The things that he doesn't like about it is that it cannot convert decimals into fraction and also it can't calculate derivative and integral.

Are the aesthetics of your calculator important? What matters most (shape, color, personalized themes, etc.)?

- He doesn't care about the aesthetic of calculator.
- Its simplicity of use, accuracy and portability have higher importance to him.

If you're using a keyboard, how would you map the keys to functions, numbers, etc. ergonomics?

- He would assign each function to a specific button.
- For inverse functions such as arcsin or nth root he would assign shift + the key for original function.
- He would make sure that his calculator has the ability to represent numbers in different formats. For instance, he would assign the key "H" for Hexadecimal and "R" for radian.

Interview Q&A - Entrepreneur

What do you use a calculator for?

- She mostly uses it to calculate the price of services and products for the clients.
- Calculate her employees' salary based on the hours they work.
- She also uses it to keep track of her personal finances.

What would you like your calculator to do? Or what is the ideal calculator for you?

- Since most of the times she uses it to calculate the amount of money, she prefers her calculator to round the amount to 2 decimals.
- She would also like to have a button to calculate the tax so it would save her time and prevent making mistakes.

What kinds of calculators have you used? (physical, apps, online, etc.). Hardware or Software?

- She generally uses a physical calculator because she is used to it.
- When she does not have her calculator with her, she uses apps and/or online calculators.

What functions do you use most often? Which ones do you use the least?

- Most often she uses multiplication, addition, subtraction, division and percentage.
- She rarely uses other functions such as sin, cos or log.

What features did you like most about your calculator. What do you not like about your calculator?

- She likes that its screen is large in size and that it has soft buttons.
- Its simplicity to work with.
- Since it is rather large, it is not very portable. She can only use it in her work place.

Are the aesthetics of your calculator important? What matters most (shape, color, personalized themes, etc.)?

- Not that much. As long as it has a decent look and is easy to work with, it would be OK.

If you're using a keyboard, how would you map the keys to functions, numbers, etc. ergonomics?

- She would map the numbers and basic functions to their associated keys in keyboard.
- She would also like to have some shortcut keys for instance "t" for calculating tax and "s" for total sum.
- She believes it would be interesting to have a calculator that can be customized by the user. For example since the tax rates, products price and employees' salaries are subject to change, it would be cool if it had a shortcut for each of them to be able to modify the amount when the change happens.

5.3 Appendix C - Personas

Sarah Garrell (20)

Job Title: 1st year HR student
Education: High School + CEGEP
Experience:

- Starbucks Barista
- Summer camp counselor

Goals:

- Get her degree and work in recruiting
- Pass accounting and finance

Goals and Tasks user accomplishes
Mostly she is worried about her finance class so anything that would help her with that would be appreciated.

Problem calculator solves
She needs a calculator to calculate the equations for finance class. Her school does not allow her a programmable calculator so she will need to memorize the equations. She would definitely appreciate a it if she could enter the equation from left to right just like she memorized them.

Kevin Donnavan (23)

Job Title: Engineering Student
Education: 3rd Year Mechanical Engineering
Experience:

- 3rd Year Mechanical Engineering
- Summer internship as a junior structural engineer
- Army reserves - Infantry

Skills:

- Problem Solving, Mathematics
- Programming in Java, C#, and C++

Goals:

- Obtain a good GPA and find a job in his field.

Goals and Tasks user accomplishes
Kevin says he just wants to get through his classes and get a decent GPA. Like everyone, his hardest classes mostly have to do with math (although he feels he is better than average). Kevin will be happy with anything that can make his math calculations easier.

Problem calculator solves
The calculator helps Kevin get fast answers to difficult math problems he sees in class. Without a calculator, he is not sure how he would calculate the various functions that he sees on a daily basis. The calculator has to be precise enough so he can get the right answer to complex solutions of differential equations but he is not willing to wait - calculation must be near-instantaneous.

Tarek Ghamzi (23)

Job Title: Engineering Student
Job Title: 3rd year Electrical engineering student
Education: High School + CEGEP, currently in Electrical Engineering
Experience:

- Subway
- Pharmacist assistant

Skills:

- Problem Solving, Mathematics
- Programming in C++ , and arduino

Goals:

- Finish his degree with a good GPA
- Find a job in his field

Goals and Tasks user accomplishes
Tarek claims that his main priority in life at the moment is to get his degree in Electrical Engineering. He claims that his field is heavily based on math, which he struggles with. He aims to graduate with a higher than average GPA to gain an edge over others in his highly competitive field.

Problem calculator solves
His calculator helps him in computing the high level mathematical functions that would take hours to solve by himself. It also helps him double check his answers for simple calculations. Tarek claims that his calculator is with him at all times. Its accuracy, speed and comfort are of highest value to him.

Arash Mohajer (28)

Job Title: 2nd year Avionics Engineering student
Education: High School + CEGEP, currently in University
Experience:

- Completed two internships at a company that manufactures Flight Simulators
- Worked part time as a waiter

Skills:

- Mathematics, Physics
- Technical Writing
- Some experience programming in C# and Java

Goals:

- Find more internships during his degree
- Finish his degree in a reasonable amount of time
- Save money for his future (manage personal finance)

Goals and Tasks user accomplishes
Arash wants to finish his degree in Avionics as soon as possible so that he can get a good job in a field that he enjoys. He wants to continue taking part in engineering competitions and hackathons to learn more about his field and others and to meet other like-minded individuals. He wants to manage his personal finance in order to pay off the debt that he currently has from his university tuition.

Problem calculator solves While he is more focused on graduating than getting good grades in his classes, he has many math and physics intensive classes where he relies on a calculator. He uses a calculator at school for homework, projects and exams. He also uses a calculator for conversion between different units for engineering and physics problems. At home, he uses his calculator to manage his personal finances and to plan his future spending.

Victoria Benlolo (35)

Job Title: 1st year HR student
Education: High School + CEGEP
Experience:

- Has owned and managed a spa for 2 years
- Worked as a financial analyst for 7 years

Skills:

- Economics, Business, Finance
- Public Speaking
- Investing

Goals:

- Maximize the profit of her business
- Invest in new profitable endeavours
- She is interested in opening new spas around town once her business grows more
- Hire new employees and continue to manage the finances of her business

Goals and Tasks user accomplishes
Victoria wants to continue to grow her business and potentially start franchising her spa to open up new locations around the city. She wants to hire new talent in order to expand her finance team. Her day-to-day includes managing employees' pay, keeping inventory of products, paying bills and managing business income. Additionally, she wants to continue investing in the stock market.

Problem calculator solves At the Spa, Victoria uses a calculator to calculate all of her business expenses, profit/loss, employee salaries, etc. A reliable calculator is very important to her. She uses a calculator to plan expenses in her future business expansion plans. She also uses a calculator to keep track of how her personal accounts and investment portfolios are growing.

5.4 Appendix D - Use Case Descriptions

ID	UC 1
Name	Calculate result
Description	User wants the calculator to resolve his mathematical expression to a satisfactory level of precision near-instantaneously.
Pre-condition	• Calculator is on
Post-condition	• Calculator takes user input, parses, calculates, and arrives at a correct result. • Calculator saves the result of this operation for future use (UC 2).
Basic path	1. This use case starts with the user entering a mathematical expression into the calculator. 2. When satisfied with inputted expression user presses "=" button or "enter" on keyboard. 3. Calculator performs resolution of the arithmetic expression. 4. Calculator stores the result of the expression (UC 4).
Alternative Path	1b. User enters a letter and number in order to store a variable (UC 3). 3a. Calculator detects a syntax or arithmetic error in user's input. 1. Calculator detect the type of exception. 2. Calculator displays this exception on screen (UC 4) 3. User can clear exception and return to the offending arithmetic expression and attempt to correct the error (return to Basic Path 2).

Table 2: UC 1 - Calculate result

ID	UC 2
Name	Recover previous result
Description	User wants to recall the result of a previous calculation and be able use it in another calculation.
Pre-condition	• Calculator is on • A successful calculation has already taken place (UC 1)
Post-condition	• User sees result of previous calculation and can input it into another calculation.
Basic path	1. User presses "Ans" button which will input result of the previous calculation into the current calculation. 2. User carries on with the rest of the calculation (UC 1).
Alternative Path	1a. User can press "mem" button and see a list of previous results that can be chosen for the current calculation. 1. User presses "mem" button 2. User scrolls to the desired result 3. User presses "enter" button to insert select result into current calculation.

Table 3: UC 2 - Recover previous result

ID	UC 3
Name	Store variables
Description	User want to store values that can be recalled during calculations by referencing an alphabetical label.
Pre-condition	• Calculator is on
Post-condition	• A number is stored in the calculator's memory and is ready to be retrieved by invoking its alphabetical label. • User should be able to clear or overwrite a stored variable.
Basic path	1. This use case starts with the user entering an alphabetical label that will eventually be used to recall the stored value. 2. The user then presses "equals" to indicate that a value is to be stored under the chosen label. 3. The user then presses "enter" which tells the calculator to store the variable under the aforementioned label. 4. At any point during a calculation (UC 1), the user can evoke the value stored in a variable by entering the corresponding alphabetic character. 5. The calculator substitutes the variables's value into the calculation.
Alternative Path	1a. Clearing the variable 1. User enters the alphabetic label of the variable that requires clearing (value and label appear on display). 2. User presses "clear". 3. The calculator shows the variable is now cleared.

Table 4: UC 3 - Store variables

ID	UC 4
Name	Display Result
Description	User wants clear display of the calculation as it is being inputted. User also wants the calculator to display clear results and appropriate error messages.
Pre-condition	• Result was calculated, or • User inputs values in calculator
Post-condition	• Intermediate calculation is displayed on screen. • Final results are displayed on screen. • Error messages are displayes correctly.
Basic path	1. User turns on the calculator 1a. Welcome message is displayed. 2. User enters a calculation. The input calculation is displayed on scree as it is typed. 3. User presses "=" to calculate (UC 1). 4. The result of the calculation is displayed on screen.
Alternative Path	4a. An error occured and the appropriate message is displayed to the user. 1 User can clear the error and attempt to correct it.

Table 5: UC 4 - Calculate result

6 F - Draft Calculator User Manual

Calculator Instructions:

1. Start the calculator program
2. Read the available functions listed
3. Each function will have the number of required parameters in the brackets
4. type in the function name, along with the function parameters, all seperated by space.
5. type exit to quite the program

Function Manual

Addition (...)
SYNTAX: addition [Double ...]
PARAMETER LIMIT: infinite
RETURN TYPE: Double

Subtraction (...)
SYNTAX: subtraction [Double ...]
PARAMETER LIMIT: infinite
RETURN TYPE: Double

Multiplication (...)
SYNTAX: multiplication [Double ...]
PARAMETER LIMIT: infinite
DESCRIPTION: Multiplies the all the numbers in the parameters list
RETURN TYPE: Double

Division (...)
SYNTAX: division [Double ...]
PARAMETER LIMIT: infinite
RETURN TYPE: Double
DESCRIPTION: Divides the first parameters by the 2nd. The answers of that division is divided by the 3rd, which is then devided by the 4th...
RETURN TYPE: Double

root (2)
SYNTAX: root [Double] [Integer]
PARAMETER LIMIT: 2
DESCRIPTION: Calculates the nth root of the 1st parameter, where n is the 2nd parameter
RETURN TYPE: Double

factorial (1)
SYNTAX: factorial [Long/Integer]
PARAMETER LIMIT: 1
DESCRIPTION: Calculates factorial of the 1st parameter
RETURN TYPE: Long

ln (1)
SYNTAX: ln [Double]
PARAMETER LIMIT: 1
DESCRIPTION: Calculates natural logarithm of the 1st parameter
RETURN TYPE: Double

e^x (1)
SYNTAX: $e^{[}Double]$
PARAMETER LIMIT: 1
DESCRIPTION: Calculates Eulers number to the power of 1st parameter
RETURN TYPE: Double

power (2)
SYNTAX: power [Double, Double]
PARAMETER LIMIT: 2
DESCRIPTION: Calculates parameter 1 to the power of parameter 2
RETURN TYPE: Double

Glossary

Discord A communication software hosted on the web used for scheduling, discussion and sharing of files.

Eclipse A Java Integrated Development Environment.

Git An open sourece distributed version control software.

GitHub A web service that hosts git repositories for ease of use between developers.

IntelliJ A Java Integrated Development Environment.

Java A general purpose object-oriented programming language.

JavaDoc A code documentation generator for Java.

JavaFX A graphical user interface library for Java.

Junit A library in Java used for writing unit tests.

Latex A textual interface for writing technical and scientific documents.

Transcendental Function "A function that does not satisfy any single-variable polynomial equation whose coefficients are themselves roots of polynomials".