

# COMP 354 Iteration 1 Eternity Project

Team I

Concordia University

Friday June 5, 2020

# Overview

1 Team

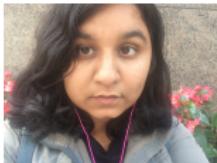
2 Requirement Gathering

3 Product

# The Team



Avnish:  $\sinh(x)$



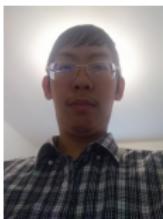
Swati:  $e^x$



Emanuel:  $\log_{10}(x)$



Clément:  $10^x$



Hong Phuc: MAD



Tara:  $x^y$



Anik:  $\sin(x)$

# Team Roles

Name	Primary Role and Responsibility	Secondary Role
Hong Phuc	GUI and Web application, prototyping	Implementation of new technologies
Swati	Organizing and planning agenda for team meetings	High-level vision, scope management
Anik	Set up of initial repository GUI for local application	Questions regarding the Python ecosystem
Avnish	Technical Writer	Latex
Clément	Major presenter, quality control	Best practices, PEP documentation
Tara	Minor presenter, Team liaison	Ensuring requirements are followed
Emanuel	Technical writer, subject matter expert for math algorithms	Algorithm optimization

# Collaboration patterns [1]: Project management

- Regular meetings once a week
- Agenda
  - Meeting transcriber ensures that discussions are on track
- Questions are gathered by the meeting transcriber
- Status Update: done, doing, to-do

# Collaboration Patterns [1]: Centralizing work

- Focus on real-time information and feedback



1

---

1

<https://discord.com/branding>

<https://icons8.com/icons/set/google-drive>

<https://github.com/logos>

# Code Reviews

- One individually assigned to each Pull Request
- Asymmetric
- Additional techniques used:
  - Inputs and discussion from all team members encouraged
  - Continuously enforce quality standards

# Gathering Requirements



2

<sup>2</sup><https://unsplash.com/s/photos/brainstorm>

# Interview Questions

- Demographic of people who use a calculator
  - Professional/educational background
  - Calculator Usage
- User needs
  - Degree of precision
  - Input box vs. button selection
  - Numeral system required
- Separate requirements into needs and preferences
  - Ease of use
  - Aesthetics
  - Features
  - Platform
- Frustrations with current device

# Interview Model

- Hourglass [2]
  - Open-ended questions
  - Followed by specific questions
  - Finished with open-ended questions
- Findings
  - Many interviewees went back to clarify a previous question answered



3

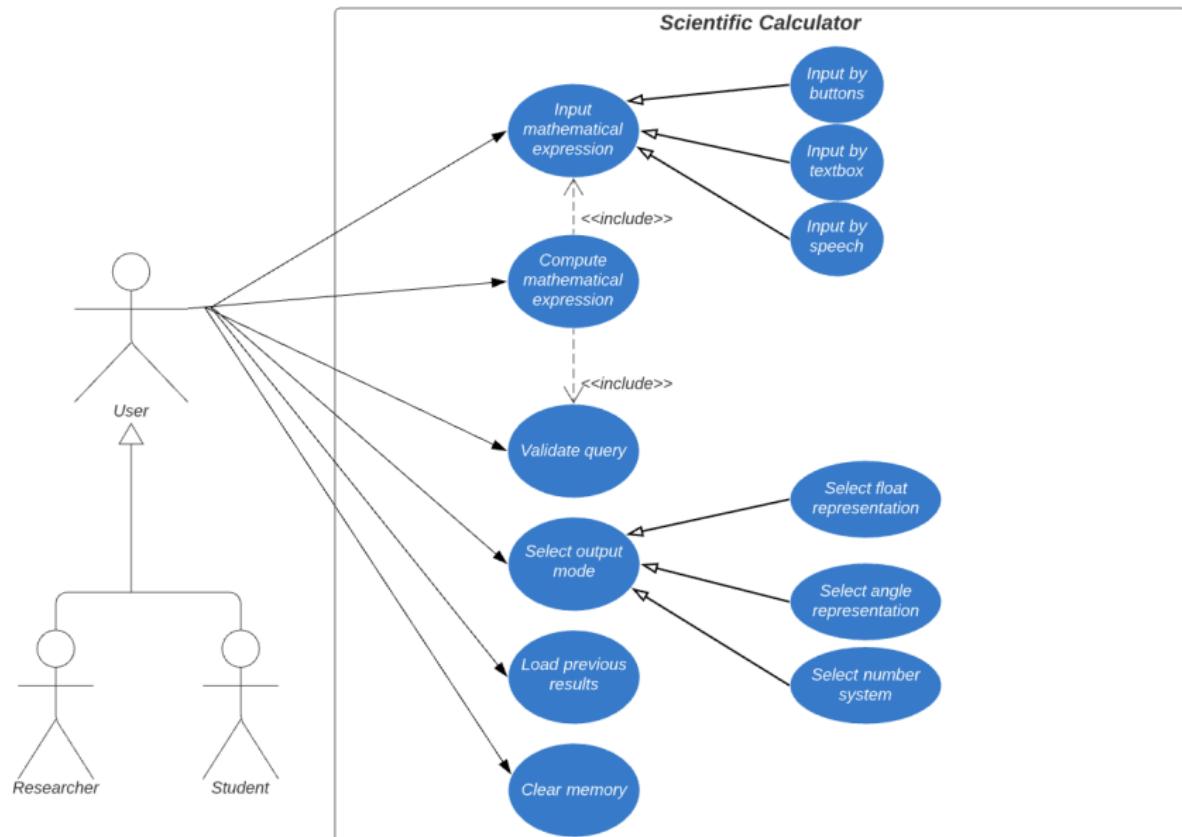
---

<sup>3</sup> <https://unsplash.com/s/photos/hourglass>

# Interview: Key findings

- Precision is extremely important
- Existing calculators contain a lot of unnecessary buttons and features
- Split findings on local or web application
  - Mobility when conducting experiments or for exams
  - Preference for desktop application when working at computer

# Summarized Use Case [3]



# Product

```
31     def __init__(self, path, debug):
32         self.file = None
33         self.fingerprints = set()
34         self.logduplicates = True
35         self.debug = debug
36         self.logger = logging.getLogger(__name__)
37         if path:
38             self.file = open(os.path.join(path, 'seen_requests'), 'a')
39             self.file.seek(0)
40             self.fingerprints.update(self.file.read().split(os.linesep))
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool('SUPERVISE_DEBUG')
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
54
55     def request_fingerprint(self, request):
56         return request_fingerprint(request)
```

4

<sup>4</sup> <https://unsplash.com/s/photos/code>

# Tech Stack

- Choice of technologies influenced by team strengths and user requirements
- Python: familiar & versatile
  - A language we were all comfortable with
  - Rapid iterations
  - Option of a GUI vs. CLI calculator
  - Native language recognition libraries
- Keeping our options open:
  - Electron for a local desktop front-end
  - Flask or nodejs for a web implementation

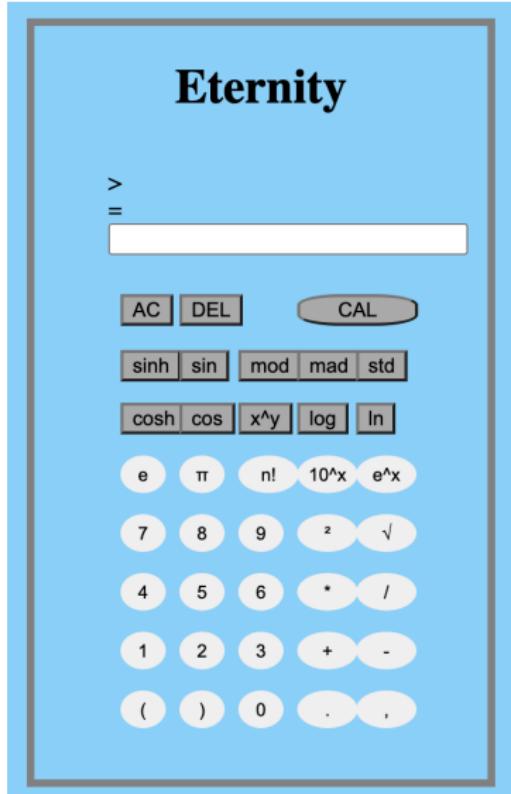
# Inclusions/Exclusions

- Information from users gave us a clear idea on what should be built
  - Some ideas were not feasible in the timeframe
    - Speech recognition
    - Radian and degree conversions
  - Scope & reasoning
    - Essential features/functionality - for iteration 1
    - Features/functionality to be looked at a later point

# Iteration 2: Organizational improvements

- Collaboration techniques
  - Buddy system
  - Role rotation
  - Better project & team wiki

# Our Product



# References

- [1] A.M. Ansiul Huq & N. Adabi, "Collaboration Patterns" presented to Class, COMP 354, Montreal, Qc., Canada, 05, 29, 2020.  
[PowerPointslides]. Available:  
[https://drive.google.com/drive/folders/1ubZfwsSh\\_mQtd0wkPU0CgCpdBdNA6ibu](https://drive.google.com/drive/folders/1ubZfwsSh_mQtd0wkPU0CgCpdBdNA6ibu), Accessed on: 06 01, 2020.
- [2] P. Kamthan, Class Lecture, Topic: "Introduction to Interviews." Concordia University, May., 2020.
- [3] P. Kamthan, Class Lecture, Topic: "Introduction to Use Case Modeling." Concordia University, May., 2020.