

# Cold Call Helper

Nolan Cassidy, Andrew Evelhoch, Jake Gianola, Kyle Kincaid, Kylie Quan

# Technologies Used

Language: Python3

GUI: Tkinter

Collaboration: Google drive

Compiler: A variety of python compilers that did not work for us on Mac

IDEs: Pycharm, Atom, Vim, IDLE



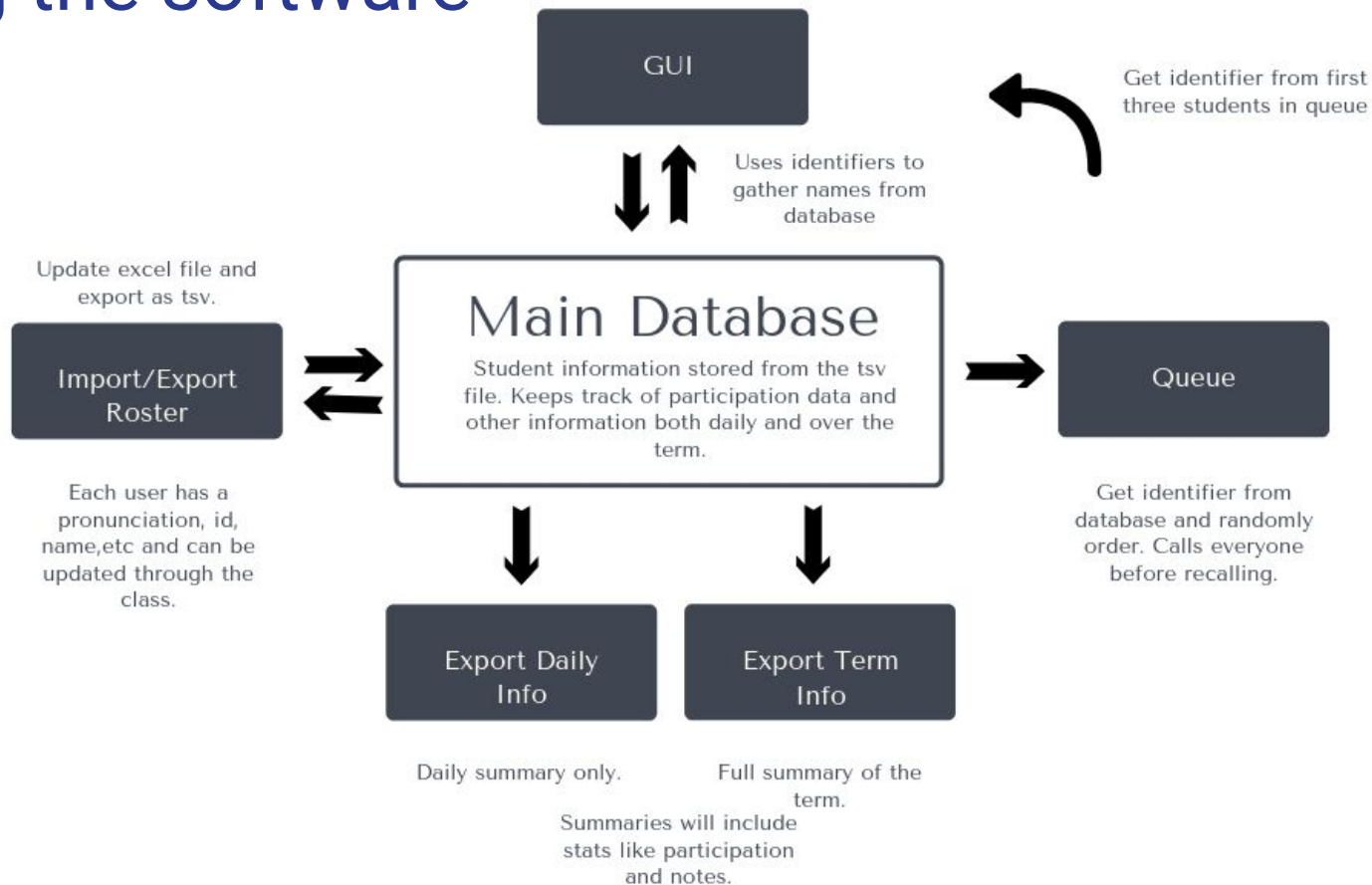
# Developing System Requirements

- Went through each bullet point of the client SRS
- The randomizer as specified in the SRS given to us had problems with it
- Experience with other teachers using similar programs in class; noticed they were not random and lacked features
- As a group we learned that requirement and analysis starts with thorough research/planning but remains a living process throughout the duration of project management.



# Designing the software

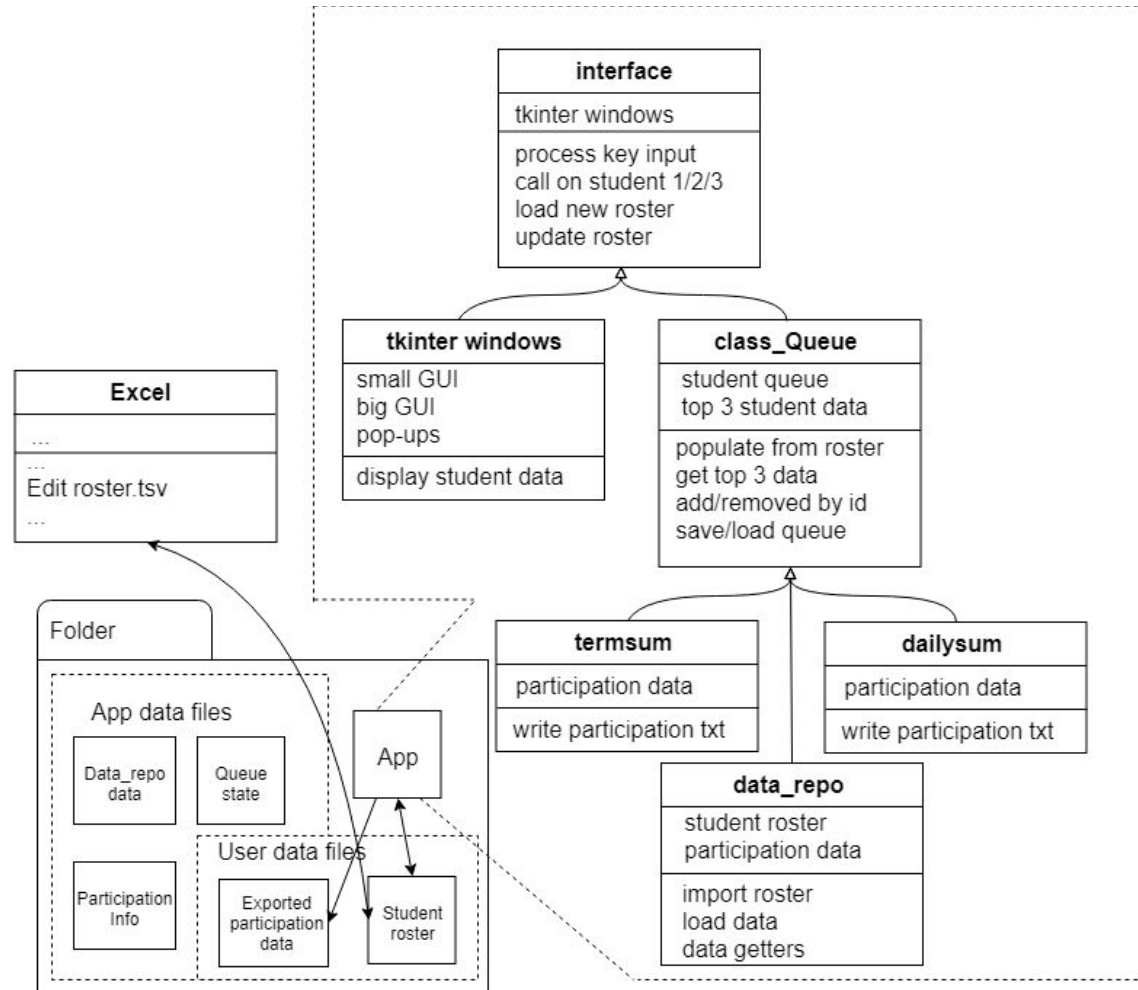
## Early design



# Final Design

-FERPA protection  
using system  
architecture

-More modular



# How we integrated


One system at a time was integrated

Began with Repository and the Queue

Difficulties due to code being worked on by multiple people

GIT repository is preferable

We learned that you must take into account cross compatibility  
earlier on in the project



# How team was managed

## **Team experience:**

- A “Divide and Conquer” strategy saved time and resources
- Each team member became a specialist at their given module

## **3 lessons we learned about software design:**

- Unexpected bugs can lead to the critical path becoming longer
  - Software design diagrams can illustrate systems in to stakeholders
  - Modules with loose coupling allow for more efficient integration
- 