# CIS 422 Project 1:
# Ducks on Deck
# User Documentation

Ellie Kobak (erk), Kalyn Koyanagi (kek), Liza Richards (ljr), and Kelly Schombert (ks)

January 30, 2022 – v2.0

## Table of Contents

## 1. User Documentation Revision History

| Date | Author | Description |
|---|---|---|
| 1-22-2022 | ljr | Created the initial document and started first draft |
| 1-24-2022 | ljr | Wrote different cases in section 2 |
| 1-26-2022 | ljr | Made changes to cases in section 2. |
| 1-29-2022 | ljr | Added photos to each section. |

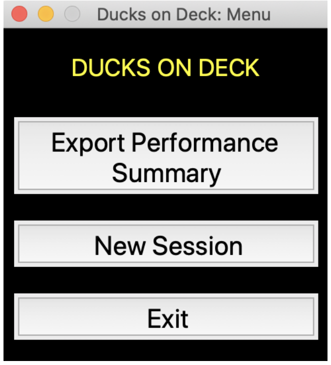
## 2. Introduction

The following sections of this document will explain all the different use cases of the Ducks On Deck software.

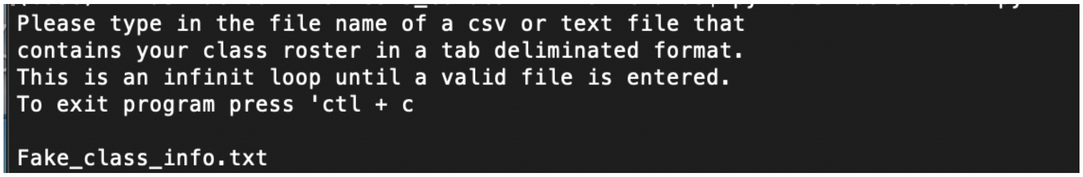
## 3. Installation

1. Go to github directory and select the Cold Call zip file
2. Download this zip file “ColdCall.zip” to your computer
3. Open the terminal application using “Command + space bar” to open the search window and type in terminal
4. In the terminal window, change the directory to where ColdCall is stored. In most cases, this command would be “cd Downloads/ColdCall

5. Once in this directory, run the software by typing “python3 DucksOnDeck.py. The following window will be displayed:

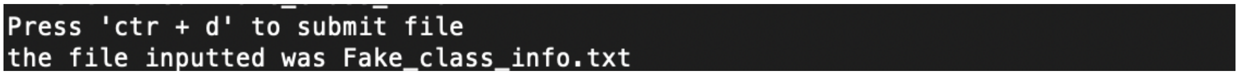


6. The terminal will prompt for user input of the desired class roster file. A test roster has been provided in the ColdCall directory. To use your own, type the path from the home directory to where your desired class roster file is located. Note that if the file is not of the .txt format, it will not be accepted by the system.



```
Please type in the file name of a csv or text file that
contains your class roster in a tab deliminated format.
This is an infinit loop until a valid file is entered.
To exit program press 'ctl + c

Fake_class_info.txt
```

7. Press “ctrl + d” to submit the file. If the file is not of the correct format, the system will not run and will instead restart.



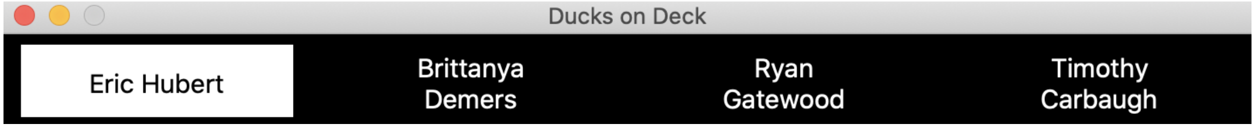
```
Press 'ctr + d' to submit file
the file inputted was Fake_class_info.txt
```

8. The system is now ready for use, select the “New Session” button on the main menu window to begin cold-calling!

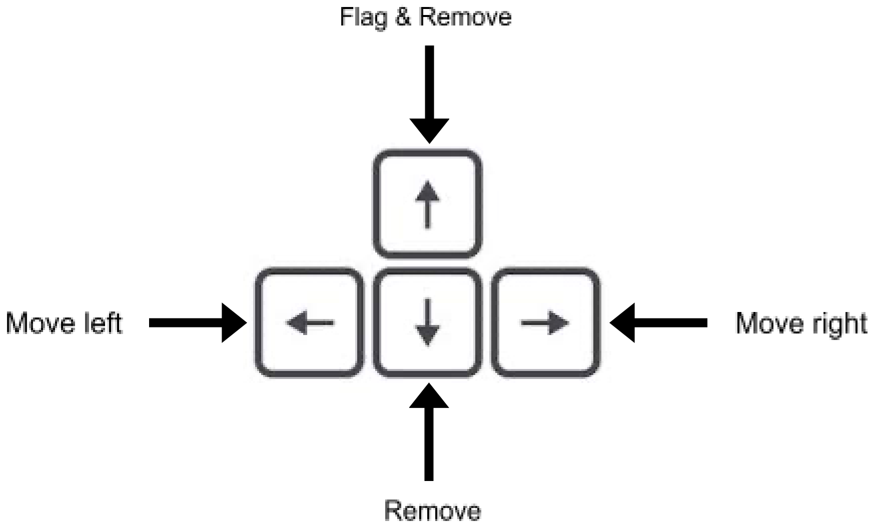
# 4. Use Cases

### 4.1 Case 1: New Session

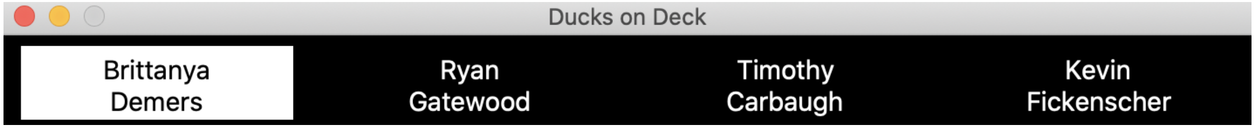
This case will lead to the “on deck” window being displayed on the user screen. This window will choose four random student names from the queue for the user to choose between. An example of this window is displayed below:



The user can then use the left and right arrow keys on their keyboard to move between the names. As the user moves between names, they will be highlighted to indicate which name you are currently on.



Once you have found the student you wish to choose, you can either flag and remove the student, or remove them from the On Deck window. To flag and remove a student, the user can press the up arrow key. This action will be recorded in a performance summary file by increasing the “Number of flags” count and the “Total times called” count for that student by 1. To only remove a student the user can press the down arrow key. This action will also be recorded on the performance summary file by only increasing the “Total times called”count by 1. After this action, the student name will be removed from the On Deck window and return to the queue. The rest of the student names will be shifted up one position, as displayed below when compared to the previous On Deck list window.



Ducks on Deck

Brittanya Demers

Ryan Gatewood

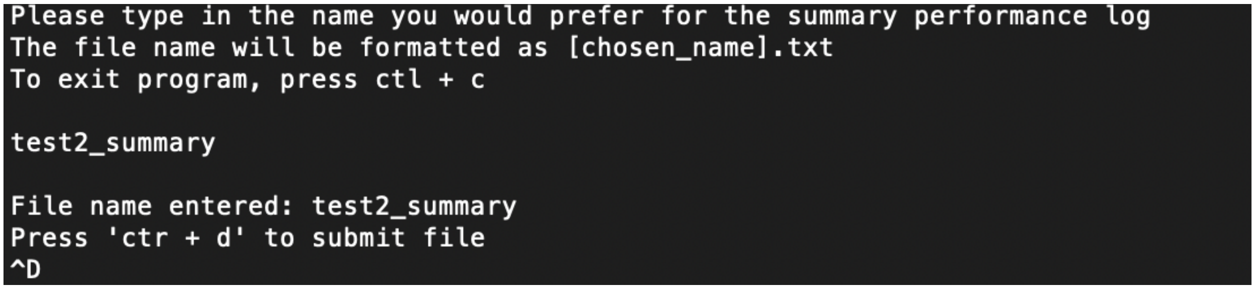
Timothy Carbaugh

Kevin Fickenscher

Notice how in the image above, after removing “Eric Hubert” from the first position in the list, the rest of the names shifted up by one position and a new student was added to the back of the “Ducks on Deck” list. To avoid quick repetition of names when a new student moves to the list, the student names will only be randomly selected from the front 75% of the queue. To stop names from being recorded, the user can press the “Exit” button, or the “Export Performance Summary” button. Note that if the Exit button is chosen before the Export Performance Summary button, any data recorded during the session will be last. To save the data, choose the Export Performance Summary button and refer to 4.2 Case 2.

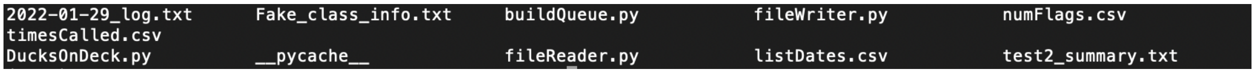
### 4.2 Case 2: Export Performance Summary

If the user selects the “Export Performance Summary” button, the user will be asked to enter the desired name of the file to be exported. This file will contain all of the cold calling information collected from the previous session. Once the name is written, press enter. It is important that the user enter the name without the “.txt” file at the end. The system will automatically save the file in a .txt format, so there is no need for the user to specify this. Then the system will repeat back to the user the name of the file they chose, and will be asked to submit the file using “ctrl + d”.



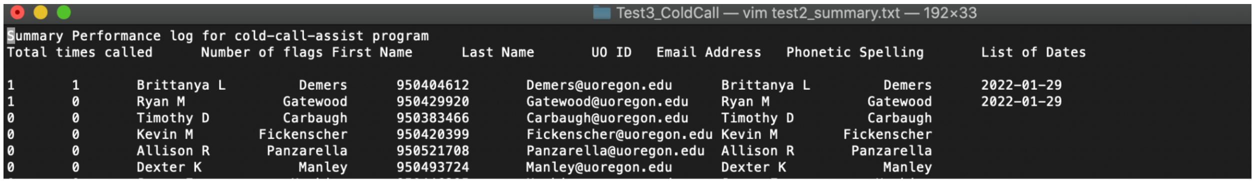
```
Please type in the name you would prefer for the summary performance log
The file name will be formatted as [chosen_name].txt
To exit program, press ctl + c

test2_summary

File name entered: test2_summary
Press 'ctr + d' to submit file
^D
```

After pressing “ctrl + d”, you must select the “Exit” button on the Ducks On Deck main menu. This button will terminate the system, and the user will be able to type into their command line in the terminal window. In the command line, you can type “ls” in the same directory to see all of the contents in the directory. You should be able to see a .txt file with the name chosen by you. As you can see below, “test2_summary.txt is now in the working ColdCall directory.



```
2022-01-29_log.txt      Fake_class_info.txt     buildQueue.py           fileWriter.py           numFlags.csv
timesCalled.csv
DucksOnDeck.py          __pycache__             fileReader.py           listDates.csv           test2_summary.txt
```

To view the contents of this file, you can type “vim <name_of_file.txt>”. This command will open the .txt file and allow the user to see its contents and make changes to it. The user can also access this file by finding where the directory folder is stored on their computer in the finder application, double clicking on it, and selecting the .txt file.



```
Summary Performance log for cold-call-assist program
Total times called      Number of flags First Name      Last Name       UO ID   Email Address   Phonetic Spelling       List of Dates

1       1       Brittanya L          Demers     950404612       Demers@uoregon.edu      Brittanya L          Demers     2022-01-29
1       0       Ryan M             Gatewood     950429920       Gatewood@uoregon.edu    Ryan M             Gatewood     2022-01-29
0       0       Timothy D          Carbaugh     950383466       Carbaugh@uoregon.edu    Timothy D          Carbaugh
0       0       Kevin M         Fickenscher     950420399       Fickenscher@uoregon.edu Kevin M         Fickenscher
0       0       Allison R        Panzarella     950521708       Panzarella@uoregon.edu  Allison R        Panzarella
0       0       Dexter K             Manley     950493724       Manley@uoregon.edu      Dexter K             Manley
```

As you can see above, this file contains the “Total times called”, “Number of flags”, and “List of Dates”. These are the values that are changed with the corresponding keystrokes while the system is running. The Total times called keeps track of the total times a student is called. The number of flags keeps track of how many times a student was flagged. The list of dates marks down which dates students were called on. To exit this window and return back to the terminal, type “esc + “:q + enter”. Now that the user is back in the terminal, they can choose to run the program again.

Now the user has a daily log of the cold calling data from the previous session. The original design of our system was that if the user would like to continue building upon this data the next time they run the system, they can save each session's data into the same .txt file. In this case, for the next session when you select "Export Performance Summary" and you are asked for a file name, you would input test2_summary again. This action was intended to expand on the already existing data in the file. However in our system there is a bug present where instead of the file being expanded upon, the exported file is instead overwritten with the new cold-calling data from the most recent session and the old data is lost. If the user would not like to build upon the already existing data and save a new session's data separately, then they can enter a different name when they are asked to input the preferred summary performance log name. Each new file name that is inputted into the system upon exporting data will create a new .txt file in the existing directory. Therefore, the user will be able to access all the performance summaries in the directory. To know which day each summary was collected on, the user can look at the "List of Dates" header in the file.

### 4.3 Case 3: Exit

When the user is done with the system, they can choose the exit option. This will completely exit the program and stop the system from running. This button will need to be pressed after typing in the file name to export the performance summary and typing in "ctrl + d".