New Cold-Call Assist Software Requirements Specification

Table of Contents

1. SRS Revision History

Date	Author	Description
2-2-2020	MM	Created the initial document.
2-3-2020	MM	Revised initial document.

2. The Concept of Operations (ConOps)

2.1. Current System or Situation

The Initial Cold-Call Assist Software Requirements Specification (ICC SRS) outlines a proposed version of the software with details of possible use cases and requirements. This document communicates the particular Cold-Call Assist software system created. Section 1 of the ICC SRS summarizes the intended use of the software; “The system is intended to assist and instructor with “cold calling” on students in a classroom”. Cold Calling is a method used by professors to randomly assign four students to answer the next set of questions.

2.2. Justification for a New System

Section 2 of the ICC SRS describes the motivation behind creating this software. Research on class participation and non-voluntary class participation has shown that students benefit from “human contact” in an educational setting.

2.3. Operational Features of the Proposed System

Cold calling is proven to be more beneficial for student participation than other call and response techniques. The proposed software system makes cold calling easier for the user. Users will be able to generate random student names at an equal probability. Guaranteed randomness of a cold calling system helps to ensure that each student gets a chance to experience the benefits of participation. Also, the software system assists the teacher in evaluating participation frequency. The Cold-Call Assist software generally makes cold calling on students faster, more reliable, and more accurate.

2.4. User Classes

User classes are groups of users that are grouped together according to their interaction with the proposed software system. For the cold calling software there are two main user classes that group together people that use the software in a similar way. User classes include Instructors and Software Developers.

Instructors
Instructors are the main users of the Cold-Call Assist software system. In a classroom setting instructors will open the Cold-Call Assist software to employ the cold call method. After calling on students, the instructors will then use the generated files (Daily Log and Summary Performance) to follow up with students and assign participation grades.

2.5. Modes of Operation

Modes of operation define the different ways that user classes interact with the software system. The modes of operation include Development Mode, Day-to-Day Mode, Daily Log Mode, Student Roster Mode and Summary Performance Mode. These modes are all derived from the use case descriptions in ICC SRS Section 3.

1. Development Mode is related to the software development user class. This is when the software is being designed, implemented, maintained and managed.
2. Day-to-Day Mode describes daily classroom use. ICC SRS section 3A outlines the day-to-day use case.
3. Daily Log Mode is when the instructor exports the Daily Log file to follow up with a student after class. The instructor then copies and pastes the students email address and sends them an email regarding what was discussed in class. ICC SRS section 3B describes this mode in greater detail.
4. Student Roster Mode in ICC SRS section 3C details the adjustment of the student roster file halfway into the term. The instructor is in the student roster mode when importing a new student roster file.
5. Summary Performance Mode is enacted when the instructor exports to the summary performance file. The instructor can then use this information to assign participation grades at the end of the term.

2.6. Operational Scenarios (Also Known as "Use Cases")

The use cases defined in the current software system were directly derived from the ICC SRS. The four use cases that were developed are; 3.A "Day-to-day in-class usage", 3.B "After class, the instructor reviews if any students might benefit from encouragement", 3.C "Adjust the contents of the system a week into the term", and 3.D "At the end of the term, the instructor reviews a summary of the class participation.".

Day-to-day in class usage

This use case occurs when the professor is using the software system in class to randomly call on four students. Right and left arrow keys are used to highlight student names. The down arrow key removes a student that has been called on while the up arrow key removes and "flags" a student name.

After class, the instructor reviews if any students might benefit from encouragement

Student names that have been flagged will be written to the daily log file. This file lists the students that would benefit from extra encouragement or need to be asked a follow up question. The format of the file is tab-delimited and only includes which students were flagged for that day.

Adjust the contents of the system a week into the term

On the home page, the user can input a student roster file. Upon startup, the program read in the last data file used. If the user would like to update the student roster, he/she can input a new .txt file. The current data will be overwritten by the new data file and a warning will be displayed; “Are you sure you want to load in a new file?”.

At the end of the term, the instructor reviews a summary of the class participation

The professor wants to calculate participation grades and references the summary performance file. This file is written to every time a student is removed from the on-deck list. Removal from the list signifies being called on and is recorded in a tab-delimited file. This file includes the number of times a student was called and flagged along with the date they were called/flagged on.

3. Specific Requirements

3.1. External Interfaces (Inputs and Outputs)

3.1.1 Day-to-Day Input

The day-to-day classroom usage follows the use case outlined in ICC SRS section 3A. Four names are displayed on the “User View” page and are next to be called on.

Student Roster:

The ICC SRS section 5.4 “The user names will reside in a roster file with the following format:”, the file will include student name, UO ID, email address, and phonetic spelling and in a tab-delimited file. The professor loads this input file into the software through the “Import New Roster” tab on the Home Menu. The user selects the file and then can begin using the software.

3.1.2 Daily Log Output

The daily log usage follows the use case outline in ICC SRS section 3B. The user wants to offer extra encouragement or follow up questions to students flagged that day.

Daily log file:

When a student is removed and flagged, using the up arrow key, the daily log file will be generated using the “Export to Logs” button on the Home Menu. The daily log file can be viewed by pressing the “Daily Log File” button. The file includes the names and emails of students that were called on that day and the X indicates which students were flagged. This file is overwritten each day.

3.1.3 Adjusting Student Roster Input

The adjusting student roster use case is outlined in ICC SRS section 3C. Students are added/removed from the student roster and a new file is inputted.

Student Roster:

In this use case the student roster can be adjusted throughout the term by importing a new roster upon starting the program. The system will warn the user; "Are you sure you want to load in a new file?".

3.1.4 Summary Performance Output

The summary performance usage follows the use case outlined in ICC SRS section 3D. A term-long file is generated to track student participation.

Summary Performance File:

When a student is removed or flagged/removed signifying being called on, the information can be written to a term-long summary file. The file is exported using the "Export to Logs" button on the Home Menu and can be accessed through selecting the "Performance File" button. The format of this file is tab-delimited and includes the amount of times a student has been called on, flagged, and on what dates. These are listed along with basic student information from the student roster. The file also includes a reveal code but contrary to ICC SRS section 3D.4 the code is 0 or 1 and is not used to calculate participation scores but is used to signify which students have been flagged.

3.2. Functions

3.2.1 Day-to-Day Functions

The day-to-day classroom usage follows the use case outlined in ICC SRS section 3A. Actions are listed on the Home Menu and four names are displayed on the "User View" page.

a. The User View will hold four random student names and evaluate keystrokes (highlight, remove, flag) as described in ICC SRS. 4.1.A "Arrow Key Input".
b. The window will remain in front of other windows being used and the system will run in the background. For example, the professor can be using PowerPoint while also displaying the student names (ICC SRS 4.2 "PowerPoint is the active application, but the Cold Call window is the foreground window.").
c. The Home Menu window will vertically list each button. These buttons include: "User View", "Import New Roster", "Export to Logs", "Daily Log File", "Performance File", "Exit Program". These are created using tkinter display settings.
d. The User View will horizontally display four student names (<first><last> <first><last>) as described in ICC SRS section 4.3 "Display" using tkinter display settings.
e. When called on student names will be inserted back into the queue after the first n% as described in ICC SRS section 5.1 "The Ordering of Students in the Queue". The n parameter is set in the source code.
f. The queue is reloaded from the previous data file each time the program is restarted.

3.2.2 Daily Log Functions

The daily log usage follows the use case outline in ICC SRS section 3B. The user wants to export a summary of which students were flagged that day.

a. The daily log file will be exported each day through the “Export to Logs” button and displayed through the “Daily Log” button.
b. The file path for the daily log file is Cold Call.app/Contents.

3.2.3 Adjusting Student Roster Functions

The adjusting student roster use case is outlined in ICC SRS section 3C. The user wants to update the student roster tab-delimited file.

a. ICC SRS section 3C, a new student roster can be imported at any point in the term without removing current student data. The imported roster will add to the current roster, or edit student information currently in the roster.
b. “Import New Roster” on the Home Menu will be selected and a screen to select a file from the user’s local system will appear.
c. The file must be .txt format otherwise a file type error will say “Unable to open file” and “File must be a text file”.
d. The first line of the roster file will describe the columns. For example, line 1 of sample_data.txt is, “Comments: FNAME LNAME UOID EMAIL PHONETIC REVEAL NUMCALLED(opt) NUMFLAGGED(opt) DATES(opt)[date1 date2 ...], separated by tabs.
e. In accordance with ICC SRS section 3C.3, a warning will be displayed asking the user if he/she wants to import a new file.

3.2.4 Summary Performance Functions

The summary performance usage follows the use case outlined in ICC SRS section 3D. The user wants to export and track information from the whole term.

a. The summary performance file will be written to each time the user selects “Export to Logs” on the Home Menu.
b. The summary performance file will pop up after selecting “Performance File”.
c. The file path for the summary performance file is Cold Call.app/Contents.
d. ICC SRS section 3D.3 outlines the format: <total times called> <number of flags> <first name> <last name> <UO ID> <email address> <phonetic spelling> <reveal code><list of dates>. All of these fields are separated by tabs. This exact format is used for the summary performance file.
e. The reveal code is listed as 0 or 1 in the summary performance file because it will not be implemented as discussed in ICC SRS section 3D to compute the cold call participation score. Instead the reveal code is 1 if the student has been flagged that day and 0 if not.
f. The formatting of dates in ICC SRS section 3D.3 will be followed listed last in the row of student information as [DD/MM/YY’] in chronological order.

3.3. Usability and Performance Requirements

3.3.1 Day-to-Day Usability and Performance

The day-to-day classroom usage follows the use case outlined in ICC SRS section 3A. Four names are displayed on the “User View” page and are next to be called on.

a. ICC SRS section 6.1, within one second of clicking on the app icon the system will startup. It also does not take more than 10 steps or 30 minutes for an experienced programmer to compile and run the software system.
b. The system follows all appropriate build specifications listed in ICC SRS section 8. Those including Mac OS and python are adhered to.
c. The user will input a roster file the first time running the program and not have to re-input a roster each time. (ICC SRS 6.1.A) The program reads from the last data file imported.
d. The queue is generated in less than one second.
e. The system will not produce any errors that will interfere with the running of the program (ICC SRS section 6.1)
f. Keystroke actions will not interfere with the program and produce the correct output (remove, flag, highlight) and are enacted immediately within less than one second.
g. The algorithm for choosing and replacing students in the queue follows specifications listed in ICC SRS section 5.1 “The Ordering of Students in the Queue”. The parameter n is calculated in the source code and students are chosen from the front n% of the queue and placed at the end upon removal. This algorithm avoids repetition across multiple days.

3.3.2 Daily Log Usability and Performance

The daily log usage follows the use case outline in ICC SRS section 3B. The user wants to offer extra encouragement or follow up questions to students flagged that day.

a. The user is able to select “Export to Logs” before exiting the program and view the daily log file by pressing the “Daily Log” button on the Home Menu.
b. ICC SRS section 6.1.c. if the daily log file cannot be found a warning will display “Unable to open Daily Log File, please press “Export to Logs” to generate the file”.

3.3.3 Adjusting Student Roster Usability and Performance

The adjusting student roster use case is outlined in ICC SRS section 3C. Students are added/removed from the student queue through importing student rosters.

a. Student roster information is added to the queue and edited/changed student information is overwritten while new information is added.
b. Warnings as discussed will be displayed if the file is not a .txt file or if the “User View” is selected without first “Import New Roster”.
c. The program will read from the last imported data file if a new roster is not provided (ICC SRS section 6.1.a).

3.3.4 Summary Performance Usability and Performance

The summary performance usage follows the use case outlined in ICC SRS section 3D. The user wants to efficiently export a file including term-long information.

a. ICC SRS section 6.1.c. if the daily log file cannot be found a warning will display “Unable to open Daily Log File, please press “Export to Logs” to generate the file”.

3.4. Software System Attributes

3.3.1 Day-to-Day Software System Attributes

The day-to-day classroom usage follows the use case outlined in ICC SRS section 3A. System attributes such as security, and usability are important in having a user friendly and safe software to use daily. Correctness of keystrokes and randomization are also crucial in providing names that do not repeat across multiple days.

a. Security: the “User View” will only display students first and last names as stated in ICC SRS section 4.3. This ensures that private student information such as UO IDs and email addresses are not displayed to the entire class. This information is strictly for the instructor.
b. Usability: the “Home Menu” provides access to every functionality provided in the software system.
c. Correctness: Each key press enacts the appropriate action and writes the correct information to each respective file (ICC SRS section 3A.7).
d. Correctness: As described in ICC SRS section 5.1, the algorithm will not place the same student in the front of the queue twice. This ensures correctness of the cold call system in that each student is not called too frequently or infrequently.

3.3.2 Daily Log Software System Attributes

The daily log usage follows the use case outline in ICC SRS section 3B. Correctness of flagged student information allows for the user to follow up with the correct student.

a. Correctness/reliability: The system correctly indicates which students are flagged using the up arrow key. The reveal codes for these students are changed to 1 and added to the daily log file.

3.3.3 Adjusting Student Roster Software System Attributes

The adjusting student roster use case is outlined in ICC SRS section 3C. The ability to edit student information ensures that if a mistake is made it can be re-written.

a. Correctness: The instructor imports a student roster file and realizes the last name of a student is spelt incorrectly. A new roster file can be imported, and the information will be edited without overwriting the entire file.

3.3.4 Summary Performance Software System Attributes

The summary performance usage follows the use case outlined in ICC SRS section 3D. Correctness/reliability of the information written to the summary performance file allows the instructor to make grading decisions based on recorded participation.

a. Correctness: The amount of times a student has been called, flagged and on what dates helps the instructor reap the full benefits of the Cold-Call Assist software.
b. If these numbers are recorded incorrectly, the software is no longer reliable.

4. References

ISO/IEC/IEEE Intl Std 29148:2011. (2011). Systems and software engineering — Life cycle processes — Requirements engineering. https://ieeexplore.ieee.org/document/6146379

van Vliet, Hans. (2008). *Software Engineering: Principles and Practice*, 3nd edition, John Wiley & Sons.

5. Acknowledgements

ISO/IEC/IEEE Intl Std 29148:2018. (2018). Systems and software engineering — Life cycle processes — Requirements engineering. https://ieeexplore.ieee.org/document/8559686

Faulk, Stuart. (2013). *Understanding Software Requirements*. https://projects.cecs.pdx.edu/attachments/download/904/Faulk_SoftwareRequirements_v4.pdf

Oracle. (2007). White Paper on "Getting Started With Use Case Modeling". Available at: https://www.oracle.com/technetwork/testcontent/gettingstartedwithusecasemodeling-133857.pdf
IEEE Std 1362-1998 (R2007). (2007). IEEE Guide for Information Technology–System Definition–Concept of Operations (ConOps) Document. https://ieeexplore.ieee.org/document/761853

IEEE Std 830-1998. (2007). IEEE Recommended Practice for Software Requirements Specifications. https://ieeexplore.ieee.org/document/720574