On Deck
Software Requirements Specification

K. Hosaka (kah), A. Huque (ash), L. Jim (lmj), M. Terry (mht) - 02-03-2020 - v2.05

Table of Contents

1. SRS Revision History

Date	Author	Description
1-12-2020	ash	Created the initial document.
1-14-2020	ash	Wrote sections 2.1- 2.6, amended sections 4 and 5
1-15-2020	mht	Wrote section 3.1 - 3.5, formatting
1-15-2020	ash	Formatting table of contents, section 2, and section 3
1-16-2020	kah	Proofreading, small edits
1-16-2020	ash	Revisions to section 2
1-17-2020	ash	Edited and formatted use cases
1-17-2020	ash	Added citation in section 4, formatting section 3, edited section 3.3
1-17-2020	lmj	Added citation for template
1-17-2020	mht	Edited sections 3.1-3.5, formatting changes
1-17-2020	mht	Split performance requirements into “required” and “not required”
2-01-2020	kah	Initial revision for final submission.
2-02-2020	kah	Detailed revision of entire SRS for final submission.
2-03-2020	kah	Second detailed revision of entire SRS for final submission.
2-03-2020	mht	Editing and checking accuracy.
2-03-2020	mht	Changed some headers.
2-03-2020	lmj	Minor edits.
2-03-2020	ash	Formatting, rewrote sentences in section 3.4.4-3.4.6
2-03-2020	mht	Deleted redundant sections and inaccuracies.
2-03-2020	lmj	Minor edits.
2-03-2020	kah	Reviewed and finalized.

2. The Concept of Operations (ConOps)

The Concept of Operations provides details regarding the system characteristics for the classroom cold-call assist software program known as *On Deck*.

2.1. Current System or Situation

As described in the Initial Cold Call SRS (ICC SRS) 1. "System Summary", cold-calling is a technique that engages all students in the classroom environment by requesting input from randomly selected students. Incorporating "warm up" techniques such as displaying the names of students that will be called on prevents students from being blindsided when called upon; thereby increasing the efficiency and effectiveness of the cold-calling technique.

2.2. Justification for a New System

As described in the the ICC SRS 2. "The Motivation" describes how some manual methods of cold calling do not accurately randomize the students selected since it allows for some level of

unintentional bias by the instructor (Eddie, 2014). Additionally, using "warm up" techniques are seldom used in classroom settings.

In an effort to increase the usage of cold calling, it is important to optimize the “warm up” technique. Additionally, if an instructor needs to follow up with a student, manual cold calling does not efficiently record data on student involvement or allow the instructor to efficiently flag a student to follow up with. This can delay or prevent instructors from noticing students who would benefit from more encouragement. With an improved system, instructors would also have the ability to view a summary at the end of the day or term.

2.3. Operational Features of the Proposed System

On Deck is largely derived from the ICC SRS, in that it is intended to assist an instructor with cold calling students in a classroom. *On Deck* displays the names of four randomly chosen students, allowing those students to "warm up" to being called on. After a student speaks, the instructor can use the left and right arrow keys to navigate to a student's name, and then dismiss or flag a student using the up and down keys. At the end of the day or at any time during the term, a daily log will be exported and the instructor has the ability to export a performance summary report.

2.4. User Classes

Instructors: *On Deck’s* intended users are instructors. Instructors are individuals who teach in a classroom setting of any age or grade level. Instructors are not expected to have any formal technical training. However, it is strongly recommended that instructors using *On Deck* have basic knowledge of how to use a keyboard, mouse, and access applications through the terminal on a Mac computer.

Software Maintainers: Software maintainers are needed to create modifications, push updates, and address any errors in *On Deck*. Software maintainers should be proficient in Python 3.7 or 3.8, have high attention to detail, and be able to respond to technical user complaints in a timely manner.

2.5. Modes of Operation

Instructor Mode: Normal usage of *On Deck* occurs in the instructor mode. In instructor mode, users see a home page which has two options: "Presentation" view and "Roster Information" view.

- The "Presentation" view displays the randomly generated four names on the instructor's screen in the <first> <last> format as described in the ICC SRS 4.3. "Display". The instructor removes and flags students who have participated using the arrow keys as described in the ICC SRS 4.1A. "Call on students and remove them from the 'on deck' list", but highlighting the names yellow at position 1 rather than inverting the text color.
- "Roster Information" view allows the instructor to import the roster and export a file with appropriate error checking as described in ICC SRS 4.4. "Roster Input and Output".

Specifically, instructors can also update the roster and export the performance summary report at any time throughout the term.

Settings Mode: Instructors can see Settings Mode from the homepage which includes two options: "Test Randomizer" and "Reset Application".

- "Test Randomizer" allows the instructor to test the randomization functionality as described in ICC SRS 5.5. "Random Distribution Verification Mode", the only difference being the user must press the "Test Randomizer" button inside the settings view instead of using a keystroke combination to initiate the test.
- "Reset Application" allows the instructor to clear the application and revert to an entirely new state. The user is warned about the effects of the reset before it is executed.

2.6. Operational Scenarios

2.6.1. Use Case A: Day-to-day in-class usage

This case describes how an instructor would utilize *On Deck* in a classroom setting to foster daily participation with flags and dismissals as described in the ICC SRS 3A. "Use Cases". However, the instructor must run the command "python3 On-Deck.py" to execute the program rather than searching their hard drive for the program.

2.6.2. Use Case B: The instructor reviews the daily log

This use case describes how an instructor accesses and interacts with the daily log to review student participation and email students as described in the ICC SRS 3B. "Use Cases".

2.6.3. Use Case C: Adjust the contents of the system a week into the term

This use case describes how an instructor updates the class roster that is stored in *On Deck* as described in the ICC SRS 3C "Use Cases".

3. Specific Requirements

The Specific Requirements provides details regarding the system structure and organization for the classroom cold-call assist software program known as *On Deck*.

3.1. External Interfaces (Inputs and Outputs)

3.1.1. Import Student Rosters - Must Have

Purpose: To load a tab-delimited file at the beginning of the term *On Deck* as described in ICC SRS 3E. "Use Cases" and at any point in the term as described in ICC SRS 3C. "Use Cases". This is required and was implemented in *On Deck* as the only means for instructors to load in and update class rosters.

- Source: The files are supplied by the instructors as a tab-delimited file (.tsv).
- Range: On the lower end, *On Deck* supports empty files. For the upper bound of the range, *On Deck* accommodates any number of students.
- Format: .tsv files are supported for both imports and exports.

3.1.2. Format of Student Rosters - Should Have

Purpose: This interface is utilized and implemented to check that the format of the given tab-delimited file is correct and raise an error as described in the ICC SRS 4.4. "Roster Input and Output". The formatting should be done as stated below, however only names are required as a bare minimum of functionality.

- Source: File will be created by instructors either through an Excel export, or through manual typing.
- Range: On the lower end, *On Deck* supports empty files. For the upper bound of the range, *On Deck* accommodates any number of students.
- Format: The format of each line in the file is given as such: <first_name> <tab> <last_name> <tab> <UO ID> <tab> <email_address> <tab> <phonetic_spelling> <tab> <reveal_code> <LF>, as stated in the ICC SRS 5.4 "The user names will reside in a roster file with the following format", but without the first line of the file representing the formatting information.

3.1.3. Export Daily Student Reports - Should Have

Purpose: Daily student reports are a feature of *On Deck* to be used by instructors to monitor participation at the end of the day as described in ICC SRS 3B. "Use Cases". The daily reports provide extra information that is useful to the instructor, though it was not entirely required to meet the bare minimum usable state.

- Source: The daily log is created every time the program is exited.
- The range and format follow what is specified in the ICC SRS 3B. "Use Cases", where a line is created for each student that was called that day, and the format is <flag> <tab> <first name> < last name> "<"<email address>">".

3.1.4. Export Term Student Reports - Should Have

Purpose: Term student reports are a feature of *On Deck* that summarizes total student participation at the end of the term as described in the ICC SRS 4D. "Use Cases",

but the student information does not include the reveal code. Also, the list of dates that students were called on is formatted as YY-MM-DD rather than YY/MM/DD in the term log. The summary provides extra information that is useful to the instructor, though it was not entirely required to meet the bare minimum usable state.

- Source: File is created through a button on the *On Deck* user interface. To export the term log, users will go from Homepage → Roster Information → Performance Summary Report.
- The range is a line created for each student and the units of measure are the dates representing the students who participated as described in the ICC SRS 3D. "Use Cases".

3.1.5. Export Randomness Checker - Could Have

Purpose: The randomness checker is implemented to allow the instructor to test the accuracy of the randomization functionality as described in ICC SRS 5.5. "Random Distribution Verification Mode." The key difference here being that the user must press the "Test Randomizer" button inside the settings view rather than using a keystroke combination. Though it is a useful tool for checking the accuracy of *On Deck*, this tool is more for testing and less for instructor use. As such, it is a nice addition but was not strictly required.

- Source: File is created through a button on the main *On Deck* user interface. Homepage → Settings → Test Randomizer.
- Range: The mock data of the students will be present in the file. Additionally, their percentages are displayed to show how often each student was called to check the distribution.
- Units of Measure: The number of times a student is called on is divided by the total number of cold-calls to output a percentage. All percentages are relatively equal.
- Format: The student data is presented in the format: <number of times called> <tab> <first name> <tab> <last name> <tab> <percentage called>.

3.1.5. Import Student Photos - Won't Have

Purpose: To include photos in the cold-call system so instructors can better interact with their students as described in the ICC SRS 7. "Optional Secondary System: A Photo-Based Name-Learning System". This was not implemented and would be an additional feature where instructors would have to upload photos and associate them with students.

- Source: Photo uploaded by the instructor.
- Range: 80x120 photos would be accepted as well as anything smaller.
- Format: only .jpg files would be accepted.

3.2. Functions

3.2.1. Importing Rosters Actions

- To import rosters into *On Deck* the user will select a button that will bring them to a file select screen. Once the instructor selects their tab-delimited file it will be parsed. The tab-delimited file can either be constructed by hand, or by typing each student's information into different columns of an Excel document and exporting it as a .tsv file.
- The parsing into an active roster with error handling is done as described in ICC SRS 4.4. "Roster Input and Output". Specifically, if any aspects of the student entry are missing (such as name, UO ID, etc…) an error message will be returned without crashing to the program. This error message will inform the user there was an issue with the formatting of their tab-delimited file and the file could not be parsed.
- Altering the file and reuploading will not lead to duplicates in the class roster.

3.2.2. Daily Log Actions

- The daily output functionality will follow that described in the ICC SRS 3A and 3B. "Use Cases", where four random students are displayed for an instructor to flag and dismiss students. Specifically:
    - The daily output log will be exported upon close of the system.

3.2.3 Performance Log Actions

- The term output functionality will follow that described in the ICC SRS 3D. "Use Cases", where instructors have the ability to review a summary of in class participation to gain insight on the students' progress throughout the term. Specifically:
    - To export the performance summary report as a tab delimited file, the instructor must go from Homepage → Roster Information → Performance Summary Report.
    - The instructor has the ability to select the desired location to save the performance summary report.
    - The queue is stepped through, writing each student's information into the tab delimited file.
- Errors in the student's data is caught by *On Deck* during the input stage and thus will not exist when generating output.

3.2.4 Presentation View Actions

- The presentation view functionality will follow that described in the ICC SRS 3A. "Use Cases", where four random students are displayed in front of lecture materials for an instructor to flag and dismiss students. Specifically:
    - The list of names will appear above lecture materials as described in the ICC SRS 4.3. "Display". However, unlike what is described in the ICC SRS 4.2. "Powerpoint is the active application but the Cold Call window is in the foreground window", *On Deck* will not be listening for keystrokes while being the background application. The instructor must click on the *On Deck* interface to make it the active application before using the keystrokes to navigate the system.
    - The daily output log will be exported upon close of the system.
    - To make sure the ordering of students is random, the queue keeps track of who has already been called on by removing those who have already been called. Once everyone has been called the queue resets

3.3. Usability Requirements

3.3.1. User Requirements

- The user requirements will mostly follow those described in the ICC SRS 8.1. "Target Platform". Users are able to run the system on Macintosh OSX 10.13 (High Sierra). Instead of OSX 10.14 (Mojave), users can run the application on OSX 10.15 (Catalina). Other versions or platformas may not be compatible.
- Excel is required in order to generate tab-delimited files. Additionally, the output file which summarizes student data can be imported back into Excel.

3.3.2. Program Requirements

- Functionality for instructors to input the class rosters is needed for *On Deck* to execute as expected. As described in the ICC SRS 5.1. "The Ordering of Students in the Queue", four students can continuously be randomly generated and inserted in the queue. However, the randomness of the queue is not conditional on an *n* parameter. Instead, once a student is called on they are removed from the active queue and added to a passive queue. Once the active queue has been almost entirely emptied (fewer than 4 students), the passive queue is randomized and put back into the active queue. The queue can be reloaded with both the active and passive rosters preserved.
- While the export system that reports the results of *On Deck* is ideal, it was not crucial to the functionality of the program. Other components of the program, such as numeric input for selecting student names, photos, or the exportation of the randomness verification document, could all be left out of the prototypical state and are not essential to the basis of what is required to general cold-calls for a class.

3.4. Performance Requirements

3.4.1 Program Startup Requirements

Required and Implemented in On Deck

- Program opens to the most recently opened class roster in less than 5 seconds.
- 4 students are automatically placed on deck upon program opening:
    - Note: the instructor will need to import a class list before they can use *On Deck*.

Not Required and Implemented in On Deck

- As stated in ICC SRS 5.1. "The Ordering of Students in the Queue", if the program was closed while the student queue was active, a saved state instance of *On Deck* will be loaded in for the instructor. This means the “on deck” students from before closing the program will be “on deck” upon re-running.

3.4.2 File Input Requirements

Required and Implemented in On Deck

- The software parses files with any number of entries.
- The software returns error messages for empty tab-delimited roster files.
- The class roster, as inputted by the instructor, is parsed and available for use within 5 seconds.
    - During this time the information is also stored in a queue.

Not Required and Not Implemented in On Deck

- Class roster backup is saved separately from the input file so the instructor is not entirely dependent on keeping the same file in the same location.

3.4.3 Use Requirements

Required and Implemented in On Deck

- Key inputs are registered instantly (<1 second).
- Removal of on deck students with a new student appearing on the list takes less than 1 second.
- The queue maintains a random order with queue resets taking less than 3 seconds.

Not Required and Implemented in On Deck

- As stated in ICC SRS 4.1. "Call on students and remove them from the 'on deck' list", navigating over the names of students will highlight it in less than a second. Rather than inverting the text color, however, there will be a yellow highlight at position 1.
    - The cursor starts on the left most name. The instructor can use the left and right arrow keys to navigate between any of the names that are “on deck.”

- The instructor can press the “up” key to dismiss a student. The instructor can press the “down” key to dismiss and flag a student.

3.4.4 Output Student Performance Requirements

Required and Implemented in On Deck

- A new file is created to write to in less than 1 second.
- 100% of the student’s information will be available on the file within 10 seconds.
- The output parser is able to write any number of lines, depending on how many lines are in the input.

3.4.5 Output Randomness Verification Requirements

Required and Implemented in On Deck

- Program has randomization functionality to randomly test cold-calling 1000 students as described in ICC SRS 5.5. "Random Distribution Verification Mode.” In contrast to ICC SRS 5.5, we do not restart instances of the queue. Rather the test randomly removes 1000 students and tracks how often each was called.
- The output file containing the results of the verification test is generated in less than 10 seconds.

Not Required and Not Implemented in On Deck

- Users should be able to quit anytime before 1000 cases are completed as described in ICC SRS 5.5. "Random Distribution Verification Mode".

3.4.6 Save State and Shutdown Requirements

Required and Implemented in On Deck

- Upon executing the program, the state of the queue is saved as stated in the ICC SRS 5.1. "The Ordering of the Students in the Queue".
- In the shutdown process, a daily log is written to reflect the performance of students that got cold-called. This takes less than 3 seconds.

Not Required and Implemented in On Deck

- Daily logs are written to the same file without creating a new one every time.
- Daily log automatically opens within 10 seconds of exiting the program.

3.5. Software System Attributes

3.5.1. Usability

Because *On Deck* will primarily be utilized by instructors, it is essential that the software is intuitive and easy to use. Attributes of the software, such as the user interface, are very

basic. Once deployed to instructors, it is important that visual aspects of the program do not change dramatically between versions. If there are changes, more documentation will be required. Additionally, extensive documentation surrounding the use of the program is included to promote reusability among instructors.

3.5.2. Integrity

Instructors will be inputting a large number of their students into *On Deck*. As such, we want each student's data (ID number, email, photo, etc…) to be secure. This will largely be done by taking the program offline and storing information on local machines. However, this will limit reusability between machines as information will not be passed from one computer to another.

3.5.3. Correctness

It is imperative that each student that is inputted into the system is included in cold-calling. This is done through a queue system. Once a student gets called on, their name will be randomly placed in a passive queue that becomes active once almost all of the other names get called on. In this manner, all students will be called upon and the order of the students will always be random. This places constraints on testability due to the randomness of the operation.

4. References

Dallimore et al. (2006). Nonvoluntary Class Participation in Graduate Discussion Courses: Effects of Grading and Cold Calling. Journal of Management Education, 30(2), 354-377, https://journals.sagepub.com/doi/abs/10.1177/1052562905277031.

Eddy, S. L., Brownell, S. E., & Wenderoth, M. P. (2014). Gender gaps in achievement and participation in multiple introductory biology classrooms. CBE life sciences education, 13(3), 478–492. doi:10.1187/cbe.13-10-0204

Knight, J. K., Wise, S. B., & Sieke, S. (2016). Group Random Call Can Positively Affect Student In-Class Clicker Discussions. CBE life sciences education, 15(4), ar56. doi:10.1187/cbe.16-02-0109

Hornof, Anthony. (2019). CIS 422 Software Requirements Specification Template. Downloaded from https://classes.cs.uoregon.edu/20W/cis422/Handouts/Template-SRS.pdf in 2020.

Hornof, Anthony. (2020). CIS 422 Software Requirement Specification (SRS) for a Classroom Cold-Call Assist Software Program. Downloaded from https://classes.cs.uoregon.edu/20W/cis422/Handouts/Cold_Call_System_SRS_v2.pdf in 2020.

“Active Learning.” *Derek Bok Center, Harvard University*, bokcenter.harvard.edu/active-learning.

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

5. Acknowledgements

This template builds slightly on a similar document produced by Stuart Faulk in 2017, and heavily on the publications cited within the document, such as IEEE Std 1362-1998 and ISO/IEC/IEEE Intl Std 29148:2018.

This document uses a template provided by Anothony Hornof in CIS 422 at the University of Oregon. These templates are "<Project Name> Software Requirements Specification [Template]" and "How to Develop a New SRS Based on the Initial Cold-Call-Assist SRS". The ideas contained in this document were generated from his "Software Requirement Specification for a Classroom Cold-Call Assist Software Program".