CoolCall: A Cold-Call Assist Program
Project Plan

Arden Butterfield, Quinn Fetrow, Derek Martin, Amy Reichhold, Madison Werries

January 28, 2022

Table of Contents

The structure and format of this document was derived from and inspired by the Software Design Specification Template noted in the References section of this document, which was created by A. Hornof, 2019

1. Project Plan Revision History

Date	Author(s)	Description
1-6-2022	Madison	Created the initial document based on A. Hornof's template
1-11-2022	All	Wrote initial proposal
1-28-2022	Madison	Started final write-up
1-30-2022	Arden	Finished final write-up

2. Team Organization: Group Roles

After discussing our group's strengths and weaknesses, each member was assigned a general role for the project. These assigned roles are considered to be guidelines rather than constraints, leaving room for members to request help on larger or more complex portions of the project as needed. The group roles were assigned as follows:

- Arden Butterfield - UI Designer
- Quinn Fetrow - Project Organizer
- Amy Reichold - Software Developer
- Derek Martin - Documentation/QA Engineer
- Madison Werries - Data Engineer

3. Remote Meeting Plan

3.1. Weekly Remote Meetings Schedule

For the duration of the project, the team planned to have two remote Discord meetings each week on the following days and times:

- Mondays from 6:00 pm - 7:00 pm
- Thursdays from 4:30 pm - 5:30 pm

3.2. Remote Meeting Format

It was decided that the meetings would be held online through Discord due to COVID and other work constraints. Our primary means of communication during the meetings was voice chat over Discord, with the inclusion of share-screen sessions and live-editing of a Google Doc created specifically for each meeting. As the team was often editing documents while discussing, as well as sharing and viewing one another's screens, we decided against video-calling to mitigate distractions during the meeting.

3.3. Meeting Agendas

While not concretely, this section discusses the typical meeting agenda which includes the following:

1. A brief period to chat, settle in, and greet one another.
2. The creation of a shared Google Doc to jot down notes about the meeting, including what was discussed and decided on. The primary editor of this document was chosen on a volunteer basis.
3. Time for each person to give a brief summary of their progress since the last meeting.
4. One or more code reviews, wherein team members would share their screens one at a time through Discord, walking other team members through the lines of code which have been added or modified.
5. Time for comments, questions, and concerns after each group member had shared their updates.
6. A brainstorming discussion about what still needed to be done for the project, which was recorded in the meeting's shared Google Doc.
7. A discussion of what needed to be done before the next meeting.
8. Individual work assignments for the items to be completed. These work assignments were decided based on considering each team member's thoughts and preferences, allowing team members to volunteer for tasks they were most excited about. After this, any remaining tasks were assigned based on a mutual discussion of roughly equal amounts of work per-person. The methods for deciding work designations is discussed further in section 3.4.
9. A verification from all group members of the next meeting time, rescheduling based on group availability if necessary.

3.4. Work Designations

Before each group meeting begins and the meeting's shared Google Document is created, the team used the designated Discord text chatroom to create a brainstormed list of agenda items to discuss, including any questions or concerns group members had.

By the end of each meeting, all group members are assigned a list of near-term tasks, established deadlines, and/or specific deliverables to produce. While each task will have one person who is responsible for the core of a deliverable, most of our task responsibilities overlap with other group members so that we can effectively collaborate together.

It is important to us that all group members have the option to choose their assigned tasks based on their strengths, preferences, or group roles. This provides extra motivation for our members and sets us up to succeed in producing high quality work. Group members may volunteer to complete specific pieces of the project based on their own strengths or preferences.

3.5. Team Decision Making

The positive and welcoming culture in our group allows us to enjoy working on this project as a team. We commit to promoting inclusion by considering every idea that is prompted during each meeting, because we firmly believe that all group member's contributions are equally valuable in meeting our end goal. This includes leaving space and time for group members to form their thoughts and ideas before moving on to a discussion. Our decision-making process involves a review of each idea followed by a discussion of pros and cons until the entire team has agreed on a solution.

4. In-Person Work Sessions

Although Discord is a powerful tool for collaborating, we found that working together in-person was extremely valuable. We had three in-person work sessions throughout the project, where we booked a room in the library, and spent the better part of a day working on the project together. This was a really useful way to bounce ideas off of each other, and be able to turn to our other group mates for help without delay.

Our first work session was devoted to coding. We wrote much of the code for our project in this work session. The ease of communication from all being in the same room meant that we had virtually no challenges integrating the modules of our project together.

Our next two work sessions were devoted to writing the project documents. Again, being able to ask questions to group members and get immediate answers made this stage much more efficient, since writing these documents required synthesizing information from many parts of the project, written by multiple people.

5. Work Breakdown/Project Schedule

We tracked project milestones in a spreadsheet, keeping track of our goals for when the milestones would be met, when they were actually met, and who completed them. Printouts of the spreadsheet at two points during the project are attached at the bottom of this document.

6. Monitoring and Reporting Plan

General communication is done through Discord, where group members discuss the aspects of the project they are currently developing. A separate channel exists for a running list of To-Do items which is monitored throughout development.

We created a shared GitHub repository to facilitate the development of our code. Through our repository's commit history, we were able to trace changes to our code from start to finish.

All group members managed a shared timetable where we recorded the date and time of our contributions to the project. The original plan was to use online task-tracking through a software called Trello, but since we found we were struggling to keep up-to-date with this method, we switched to using a spreadsheet in our shared Google Docs folder, as discussed in section 5.

Our initial plan was for each group member to have another group member be designated specifically to review their work, with the following breakdown.

- Derek: Reviews Arden’s work
- Arden: Reviews Amy’s work
- Amy: Reviews Madison’s work
- Madison: Reviews Quinn’s work
- Quinn: Reviews Derek’s work

However, because of our in-person work sessions and the fact that multiple people worked on every file over its lifetime, we found that this regimented system wasn’t necessary.

7. Build Plan

7.1 General Build Plan Schedule

We began our project by individually researching and writing preliminary code for parts of the system in Python. In our first in-person work session, we took that preliminary code, and turned it into a working version of the entire system with all parts communicating. After that, we finished the remaining parts of the code that hadn’t been ironed out in that initial work session, such as parts of the roster import process and the random distribution verification mode.

7.2 Integration Plan

We used a shared GitHub repository to collaborate on code. When working all together, we kept a whiteboard where we would mark who was working on what file currently, so that no two people would work on the same file at once, to avoid any merge conflicts. When not working in-person, we used a designated Discord channel as a digital “whiteboard” to notify other project members of files being edited.

To maintain up-to-date builds across, each project member was expected to (and reliably followed through with) pushing and pulling file changes often. This resulted in what could be considered a “daily build” or “daily integration,” helping to reduce the overhead necessary for integrating the project components.

We used a system of constant integration, where we would regularly run a system test, to ensure that our changes didn't break anything. This, coupled with our in-person work sessions, ensured that our modules would work well together.

For writing the accompanying documents, we used Google Docs, which allowed us to collaborate in real time.

8. Rationale for Build Plan

We have established this general project timeline with ease of programming in mind. It is important that our group members can advance previous developments by using the functionality that we have already implemented. While discussing the structure of our queue system, we considered alternatives. For example, our group considered creating a dictionary of student objects and storing each student's identifiable dictionary key in an array. However, using a Pickle file will allow us to efficiently save entire student-related objects to disk. This helps us avoid dictionary lookups by storing entire student objects in our array.

Our initial plan for the GUI was to use Tkinter (a standard Python toolkit) for processing keyboard input. However, Tkinter may be unable to complete UI requirements such as running on top of a full screen window or taking keyboard input while in the background. To mitigate this risk, we worked on prototyping these parts of the code as early as possible so that we would have time to pursue other options if need be.

We thankfully do not have to consider requirements risks as our project specifications will remain unchanged. However, due to Covid or other personal emergencies, we must consider personal risks. We are already meeting and communicating remotely, so we can appropriately accommodate any team members who need to quarantine. There are no parts of the project that solely depend on one person's knowledge due to the overlap of our assigned tasks, therefore other team members can provide additional contributions in the event that someone is unable to complete their work.

Our group's detailed understanding of the project requirements will reduce the chance of estimation risks. We understand the extent of the project scope and are working at a pace which will allow us to complete all specified requirements well before the final deadline. If the project takes a longer time than expected, we will be able to use our buffer of available free time.

9. Project Meeting History

5.1 January 6th, 2022

First group meeting, over a Discord call. Discussed system design and architecture, made plans for collaboration, established roles, determined initial steps on the project.

5.2 January 10, 2022

Meeting over a Discord call. Considered design alternatives, and made plans for the system design. Worked on the initial project plan.

5.3 January 11, 2022

Although we didn't meet over a Discord call, we planned a time when we were all available to work together on the initial project plan, and communicated through Discord text.

5.4 January 13, 2022

Meeting over a Discord call. Discussed the initial project proposal, hashed out more specifics like internal storage of the roster and queue structures. Set deadlines, and divided up tasks.

5.5 January 15, 2022

In person work session (12 pm – 5 pm). Finalized a lot of our design decisions, and wrote much of the implementation of the CoolCall system.

5.6 January 20, 2022

Meeting with Prof. Hornof. We had a meeting beforehand to discuss questions we had, to prepare for the meeting, and to begin reviewing the comments on our initial submission.

5.7 January 23, 2022

Virtual work session. We had two Discord voice calls that day, with time in between to work individually. We tidied up the loose ends of the implementation, tested our code thoroughly, and started writing documentation.

5.8 January 26, 2022

Quinn took our code to office hours, to test that it works on the target system.

5.9 January 28, 2022

In-person work session (3 pm – 7 pm). Focused on documentation.

5.10 January 30, 2022

In-person work session (10 am – 4 pm). Focused on documentation and the final submission.

10. Breakdown of time spent per person, outside of meetings

Arden Butterfield

Arden — 01/14/2022
    10-12 Working on gui.py (since split into instructor_interaction_model.py and display.py)
Arden — 01/14/2022
    12:30-1:30 writing code for KeySequence class and random verification mode
Arden — 01/14/2022
    4:30-5:30 writing code for initial import of roster on startup
Arden — 01/16/2022
    12-5 Big group work session, focusing on the roster and main controller, but working on a lot of things
Arden — 01/17/2022
    9:30-10 Refactoring into the constants file.
Arden — 01/23/2022
    2-3, 7-8:30 testing and bug-fixing the roster and queue imports
Arden — 01/25/2022
    4-5, 9-10 documentation
Arden — 01/26/2022
    7-8:30 documentation
Arden — 01/28/2022
    3-7 Big group work session, focused on documentation
Arden — 01/30/2022
    10-4, 6-8 Big group work session, documentation and submitting

Amy Reichhold

Amy — 1/15/2022:
    8pm-2am: Going through project specification, started LogManager, looked into how to store data between runs using pickle.
Amy — 1/16/2022:

12pm-5pm: Big group work session, researched how to keep the application on top while in full screen and how to create directories using os.

Amy — 1/25/2022:

11am-1pm, 7pm-8pm: Debugged LogManager.

Amy — 1/27/2022:

10pm-12am: Created unit tests for LogManager.

Amy — 1/28/2022:

3pm-7pm: Big group work session, worked on final submission.

Amy — 1/29/2022:

8pm-2am: Worked on Programmers Documentation for final submission.

Amy — 1/30/2022:

10:30am-4pm, 6pm-8pm: Big group work session, documentation and final submission.

Madison Werries

Madison — 01/6/2022

11:00-11:30 Initial commits and document creation.

Madison — 01/12/2022

3:00-4:30 Built preliminary structures for the student class, the student queue module, and the student roster module.

Madison — 01/13/2022

4:00-5:00 Worked on the main controller, including linking the main controller to the student queue implementation.

Madison — 01/16/2022

12-5 Big group work session, focusing on implementing the student queue and saving/reading data from Pickle files.

Madison — 01/23/2022

4-6 Started working on docstrings for the Python files, classes, and functions.

Madison — 01/28/2022

3-7 Big group work session, focused on documenting the SRS, tweaked some elements of the Random Distribution Verification Mode

Madison — 01/30/2022

10-4, 6-8 Big group work session and final project meeting, documentation and submitting

Quinn Fetrow

Quinn - 01/06/2022

5:00-6:00 Created and added sample data

Quinn — 01/16/2022

12:00-5:00 Big group work session, focusing on implementing GUI and Log Manager

Quinn — 01/19/2022

11:00-4:00 Updated the interface specifications in the SDS of all modules from the initial submission

Quinn —- 01/23/2022

3:00-4:00 Renamed functions and updated the write method in Log Manager

Quinn — 01/26/2022

12:00-12:45 Tested system on Hornof's computer in Office Hours

Quinn — 01/27/2022

8:00-10:00 Reviewed Random Distribution Verification manager and proposed an updated system

Quinn — 01/28/2022

3:00-7:00 Big group work session, focused on documenting the SDS

Quinn — 01/30/2022

10-4, 6-8 Big group work session and final project meeting, finished all diagrams in the SDS and interface specifications.

Derek Martin

Derek — 01/06/2022

8:00-12:00 Created a group working document with detailed project specifications and general project notes.

Derek — 01/16/2022

12:00-5:00 Big group work session, focusing on implementing RDV mode and Display

Derek — 01/17/2022

10:00-4:00 Worked on Initial Project Plan, SRS, SDS

Derek — 01/18/2022

3:00-8:00 Revised Initial Project Plan, SRS, SDS

Derek — 1/20/2022

5-10:30 Revised source code files providing documentation and comments

Derek — 01/28/2022

3:00-7:00 Big group work session, focused on ironing out RDV Mode bugs and implementing revised output file functionality

Derek — 01/30/2022

10-4, 6-8 Big group work session and final project meeting, finished documenting all source code and reviewed SRS.

11. References

Hornof, A. (2019). CIS 422 Software Design Specification Template. Available at: https://classes.cs.uoregon.edu/22W/cis422/Handouts/Template-SDS.pdf

Group 1 Project Milestones		Last updated: January 20, 2022	Color indicates current status		
Arden Butterfield, Quinn Fetrow,					
Derek Martin, Amy Reichhold, Madison Werries					
Goal date of completion	**People working on it**	**Milestone Details/Description**	**Status**	**Date completed**	**Notes**
1-11-2022	All	Initial Project Plan	Complete	1-11-2022	
1-11-2022	All	Initial SRS	Complete	1-11-2022	
1-11-2022	All	Initial SDS	Complete	1-11-2022	
1-15-2022	Quinn	Creation of example roster file	Complete	1-6-2022	
1-17-2022	Amy, Quinn	Initial I/O - ability to read in student data from a file	Complete	1-16-2022	
1-17-2022	Amy	Creation of Python Class to store student data	Complete	1-16-2022	
1-17-2022	Madison	Basic queue to store student data with random insertion	Complete	1-16-2022	
1-19-2022	Arden, Derek	Creation of basic UI to display the student queue with highlighting	Complete	1-16-2022	
1-19-2022	Arden, Derek	Ability to manage the student queue (select, remove, flag) using keyboard input	Complete	1-16-2022	
1-19-2022	Amy, Madison	Implementation of basic daily log which stores the cold call data for each day	Complete	1-16-2022	
1-19-2022	Madison	Implementation of flagging system	Complete	1-16-2022	
	All	Integration of modules	Complete	1-16-2022	
1-19-2022	Quinn	Sit on top of a full-screen window	Complete		Struggling with how to implement using Python/Tkinter
1-19-2022	Madison	Improve queue to prevent students from being added back in too soon	Complete	1-16-2022	
1-21-2022	Derek, Quinn	Implementation of Random Distribution Verification Mode	In-progress	-	
1-23-2022	Arden	Display photos besides students names	In-progress	-	
1-21-2022	Amy, Arden	Ability to preserve data between runs	Complete	1-16-2022	
1-23-2022	All	End-of-term summary of performance system	In-progress		
1-23-2022	All	Error system for incorrect file formats, corrupted data, etc	In-progress		
1-23-2022	Madison, Derek	Testing system for RDVM	In-progress		
1-23-2022	Madison, Derek, Amy	Basic unit tests for modules	Not yet started		
1-23-2022	Arden	Make it so user can change the keybindings	Not yet started		
1-23-2022		Take input while not the active application	Halted		Struggling with how to implement using Python/Tkinter
1-27-2022		Write instructions for installation and running	Not yet started		
1-27-2022		Test system on Python 3.7, 3.8, 3.9, and 3.10	Not yet started		
1-27-2022		Test system on Mac OSX 10.13 and 10.14	Not yet started		How will we get access to these?
1-30-2022	All	Edit Project Plan/Overview according to initial feedback	In-progress		
1-30-2022	All	Edit SRS according to initial feedback	Not yet started		
1-30-2022	All	Edit SDS according to initial feedback	In-progress		(especially module diagram format/explanations)
1-23-2022	Arden	System startup/shutdown from clickable application	In-progress		
1-30-2022	All	Polish Final Submission	FINAL GOAL		

Group 1 Project Milestones

Last updated: January 30, 2022

Color indicates current status

Arden Butterfield, Quinn Fetrow,
Derek Martin, Amy Reichhold, Madison Werries

Goal date of completion	People working on it	Milestone Details/Description	Status	Date completed	Notes
1-11-2022	All	Initial Project Plan	Complete	1-11-2022	
1-11-2022	All	Initial SRS	Complete	1-11-2022	
1-11-2022	All	Initial SDS	Complete	1-11-2022	
1-15-2022	Quinn	Creation of example roster file	Complete	1-6-2022	
1-17-2022	Amy, Quinn	Initial I/O - ability to read in student data from a file	Complete	1-16-2022	
1-17-2022	Amy	Creation of Python Class to store student data	Complete	1-16-2022	
1-17-2022	Madison	Basic queue to store student data with random insertion	Complete	1-16-2022	
1-19-2022	Arden, Derek	Creation of basic UI to display the student queue with highlighting	Complete	1-16-2022	
1-19-2022	Arden, Derek	Ability to manage the student queue (select, remove, flag) using keyboard input	Complete	1-16-2022	
1-19-2022	Amy, Madison	Implementation of basic daily log which stores the cold call data for each day	Complete	1-16-2022	
1-19-2022	Madison	Implementation of flagging system	Complete	1-16-2022	
	All	Integration of modules	Complete	1-16-2022	
1-19-2022	Quinn	Sit on top of a full-screen window	Complete	1-25-2022	
1-19-2022	Madison	Improve queue to prevent students from being added back in too soon	Complete	1-16-2022	
1-21-2022	Derek, Quinn	Implementation of Random Distribution Verification Mode	Complete	1-28-2022	
1-23-2022	Arden	Display photos besides students names	Aborted	-	Decided not to implement, as it did not seem useful
1-21-2022	Amy, Arden	Ability to preserve data between runs	Complete	1-16-2022	
1-23-2022	All	End-of-term summary of performance system	Complete	1-28-2022	
1-23-2022	Arden	Error system for incorrect file formats, corrupted data, etc	Complete	1-25-2022	
1-23-2022	Madison, Derek	Testing system for RDVM	Complete	1-30-2022	
1-23-2022	Madison, Derek, Amy	Basic unit tests for modules	Complete	1-26-2022	
1-23-2022	Arden	Make it so user can change the keybindings	Complete	1-24-2022	
1-23-2022		Take input while not the active application	Aborted		Decided not to implement, as it contradicted with requirements that the powerpoint be usable
1-27-2022	Arden and others	Write instructions for installation and running	Complete	1-30-2022	
1-27-2022	All	Test system on Python 3.7, 3.8, 3.9, and 3.10	Complete	1-26-2022	
1-27-2022	Quinn	Test system on Mac OSX 10.13 and 10.14	Complete	1-26-2022	Completed during office hours
1-30-2022	All	Edit Project Plan/Overview according to initial feedback	Complete	1-30-2022	
1-30-2022	Madison and others	Edit SRS according to initial feedback	Complete	1-30-2022	
1-30-2022	All	Edit SDS according to initial feedback	Complete	1-30-2022	
1-23-2022	Arden	System startup/shutdown from clickable application	Aborted		After clarification in class, system runs from command line
1-30-2022	All	Polish Final Submission	FINAL GOAL	1-30-2022	