Cold Call Assist Software Program

Project Plan

Z. Carroll (zc) - 01-24-2022

K. Nguye (kn) - 01-24-2022

N. Onofrei (no) - 01-24-2022

L. Vandecasteele (lv) - 01-24-2022

H. Zhang (hz) - 01-24-2022

Project Plan Revision History:

Date	Author	Description
01-24-2022	hz	Created initial rewrite project plan.
01-25-2022	hz	Add headlines and infrastructure.
01-26-2022	hz	Record transfer
01-28-2022	hz	More careful division of structure
01-29-2022	hz	Final check and edit, project plan completed.

Management Plan:

Our group consists of five members; Nick Onofrei, Luke Vandecasteele, Haoran Zhang, Kenny Nguyen, and Zacree Carroll.

While the project is in progress we will each add the subcomponents we are currently working on within our assigned modules to our specific Trello board, this is done to avoid merge conflicts as well as provide an easily accessible history of who did/is doing what and when. When a specific task is complete it will be moved to the “done” section on Trello and an announcement will be made in the group discord server.

All finished work will be pushed to a private github repository, so each group member can have access to the most current versions of all the project files. Besides, if the project is too late to implement some features or there are bugs in the update, we can go back to the correct version submitted before.

A mandatory meeting will take place twice a week based on discussions on Monday's Discord, the time of the meetings is subject to change based on group member availability.

Project Build Plan:

We have three main modules: cold call operation module, import and export module and testing module. The cold call operation module and the testing module have been assigned to teams of two. The cold call operation module is assigned to hz and lv. The testing module has been assigned to kn and no. The import and export module has been assigned to one member, zc.

Division of labor

zc - Import and export student roster files, scan for correct formatting, retrieve data from imported roster files

kn - Realistic data sample generator, technical documentation

no - Testing, documentation

lv - Generation of daily log, priority queue algorithm of student

hz - User interface, keyboard match

Alternative Plan

In the current crisis-ridden living environment, in order to prevent team members from being unable to continue working due to any physical or other reasons, we have set up backup forces for each part of jobs:

zc, lv and hz backup each other.

kn and no backup each other.

Expected Schedule:

We plan to finish all the code before the 26th, so that we have enough time to deal with the problem that the implementation of some functions takes too much time, and improve the documentation part.

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Jan 2	Jan 3	Jan 4	Jan 5 Given groups and project description	Jan 6 Drive setup Discord setup	Jan 7	Jan 8
Jan 9	Jan 10	Jan 11 DUE: Project	Jan 12	Jan 13 Git setup	Jan 14	Jan 15

		plan, SRS, SDS				
Jan 16	Jan 17	Jan 18	Jan 19 user interface of cold call operation module is complete	Jan 20	Jan 21	Jan 22
Jan 23 import and export module is complete	Jan 24 testing module is complete	Jan 25	Jan 26 cold call operation module is complete	Jan 27	Jan 28	Jan 29 Final Testing
Jan 30 Turn in project						

Meeting History:

Due to COVID-19, all meetings are conducted online via Discord or Zoom.

Plenary

01-06:

Draft initial submission, distributing work.

01-09:

Initial submission final version completed.

01-13:

Design the software system and redistribute work in more detail, setting backups for each component.

01-15:

Show the first version of the user interface, and summarize the requirements from other components.

Decide to draw up the individual meeting in all of the following plenary.

Prepare all parts for work.

01-17:

Show the final version of the user interface and state work progress of each part.

01-20:

Show the import file method, and state work progress of each part.

Discuss the details of what's the format of student generator and testing procedure.

01-23:

Export file method, realistic student generator and testing procedure for the generated student list completed, discuss its reliability and whether it needs to be modified.

01-24:

Improved function display completed yesterday.

Discuss about how to re-write PP/SRS/SDS.

Import the student information in the roster into the actual cold call interface display, and discuss the reliability of adding random students to the desk algorithm.

Practice and prepare questions for the meeting tomorrow.

01-27:

Final version program running test, detecting visible errors.

01-28:

Discuss document details, assign writing work.

01-29:

Final review for all documentations, and technical file, practice presentation on Monday.

01-30:

Consolidate documents, final check and final submission.

Individually meeting

	16	21	22	23	24	25
zc	with lv,hz	with lv		with hz		
kn			with no	with no	with no,lv	
nof			wiht kn	wiht kn	with kn,lv	
lv	with zc,hz	with zc			with no,kn	with hz
hz	with lv,zc			with zc		with lv

Detail of individual meeting

01-16:

Discuss how the file functions and the remove/flag functions will be designed, and how to connect into the user interface, and the necessary change of view.

01-21:

Work on linking the import file function into view.

01-22:

Work on the student generator and testing for that.

01-23:

zc and hz work on linking the export file function into view.

no and kn retest the modified algorithm.

01-24:

Discuss how to link the student generator into the program in actuality.

01-25:

Discuss connection of priority queue algorithm with view and keys.

Milestones:

Key	zc	kn	no	lv	hz	everyone

	14	16	20	21	22	23	24	25	26	28	30
Initial version of the cold call interface											
Remove, flag function, match keyboard and display correctly in the cold call window (requires connection to actual data)											
Add model select interface, link with cold call view, final version of the user interface done											
Limit flag button can only work within 1 second after remove student											
Import file functionality											
Connect Import file function with button in view											
First version of student generator											
100 test case for student generator											

Final version of student generator, solved same name problem											
100 new test case for student generator											
Export file functionality											
Connect Export file function with button in view											
Link imported data with view											
Priority queue algorithm of student, connect to the cold-calling view interface and keys.											
Generation of daily log											
Program completed!											
Final program testing											
Turn in project											

Rationale For Build Plan:

In the Project Build Plan, we divide the implementation of the project into three parts, each part is divided into different files so that each component can be developed simultaneously and independently.

Considering the cumbersome nature of creating a large amount of student data and testing, and the particularity that the user interface, as the main body of the program, is responsible for connection in addition to construction, cold call operation module and testing module are allocated with two persons, and import and export module as a whole is allocated with one person. Then, further distribution is carried out among them to form the final work distribution.

As the main body of the whole program, the cold call operation module needs to connect various components, so the view part of this module is required to complete it as soon as possible, so that other modules can have a more intuitive understanding of its function design, but other parts of the algorithm that are more complex are less urgent. The deadline of the import and export module and testing module is set a few days before the expected completion of the project, and a certain time is reserved for component connection. The specific time is determined according to the time that team members can work.

The Milestones is the specific progress of our work. Once someone completes and uploads a part, it will be recorded here so that everyone can have an intuitive understanding of the current progress.