Usability Study

for

Group Maker

Prepared by:

Anthony Phimmasone

CPSC 430

03/19/19

Table of Contents

1. Overview of System Functionality

1a. Overview of The Document

The purpose of this document is to provide a usability study based on a user interface for the group maker software project. This usability study is used to help the implementation group create and deliver well-made software to the client. In this study, I gathered four users for a focus group that will review and evaluate a set of wireframe images of the user interface. The focus group will be asked a series of questions and provide feedback as a part of this study.

This study contains an several aspects which will be used by the implementation group to help create a well-made software: an overview of the project, three imaginary users and how they will use the final product, a paragraph describing the software's overall functions and goals, images used as a prototype for the software, feedback on the user interface from the focus group, and enhancements made to the user interface images. This document is intended to serve as a guide for the implementation team and the client as well.

1b. Purpose of The System

The purpose of the group making software is to form individual groups from an input CSV file that contains various metrics that can be used to distribute students into certain groups. Additionally, the client will be able to select options from a menu to specify how the groups should be created and what factors to include in the randomization process. The product will be a Python-based software that will create groups based on a CSV file. The product will import the CSV file, parse through it, and ultimately group students together based on the condition that the user chooses. The user will be able to sort groups based on preferred lists, black lists, personality, and gender. The user will also be able to set the number of groups once they start running the program. The program will display the groups to the user after they are created. It will also create an output CSV that will contain all the groups that have been run by the program. The software is intended to be simple and general. This is to allow multiple professors within the university to be able to use the group maker in the classroom.

1c. Client & User Characteristics and Assumptions

The client is Dr. Anewalt. She is a professor at the University of Mary Washington and teaches courses within the Computer Science department. Dr. Anewalt requested this project because she wanted an easy and fun way to create groups for her students. She wanted a system that could make these groups based off certain conditions, such as a preferred list, blacklist, personality, group size, and other factors. One assumption about the client is that she will only use the software twice per semester and per section. This is because she does not need to create groups on per se a daily, weekly, or even a monthly basis. The initial intended users of this project are all the professors in the UMW Computer Science department. Other professors have similar needs and wants as the client when it comes to creating groups for their classes. This will allow the product to be useful for the entire Computer Science department. Students will not have any access to the software. In the future this software may expand to other departments of Mary Washington, but as of now they will also not have access. An assumption about the users in general is that - like the client - they will only have to use the software a few times per semester and per section.

1d. User Interaction

There are only a few simple steps for the user (the professor) to perform when interacting with the software. The first step in creating groups is for the professor to create a CSV from a google survey form. The survey form will contain information about the students that the student will fill out. Some potential information that this form may include are the following: the students name (first and last), the student's availability on campus, the student's residents (on-campus or off-campus), the students preferred list of people to work with, the students black-list (those the student does not want to work with), personality color, etc. Once the survey form is complete, the professor will send out the form to be complete by all the students within their class. Students will fill out the form accordingly and will submit their responses to the professor. When the professor receives all the information given from the students they will download the responses of the form as a CSV file. Then the professor will open the software select the CSV file to read in. Next, the professor will select which options they want to use to create the groups (such as preferred list, blacklist, personality, group size, student residency, etc.). The software will then do all the work to generate the groups randomly. Thus, reducing any unconscious bias that could be made by the professor. When the groups are generated they will be outputted into an output CSV file within the same directory. The professor will then have to option to make any last-minute changes and finalize the CSV file before the groups are made final.

1e. References

Software Requirements Specification for Group Maker

Group Maker Requirements Document

Group Maker Project Plan Document

2. Imaginary Users

2a. Imaginary User One

Karen Anewalt is a professor at the University of Mary Washington. This semester Dr. Anewalt teaches two sections of CPSC 430 – Software Engineering. Software Engineering is a class that is both speaking and writing intensive and involves a lot of group work. In the class students cooperate and communicate to write documents with one another; as well as, collaborate to share and develop code together. Dr. Anewalt is a kind and understanding professor and wants all the students in class to get along. However, she understands that some students do not get along with other students. Therefore, she would rather create the groups based on individual students preferred lists and blacklists. Creating groups by students choosing who they want to work with and who they do not want to work with is her first main priority. Her second priority is creating the groups by the student's preference of which project they would want to work on. This is because she wants the students to be happy with the project that they receive so that they can be motivated to work on it. These two priorities are the main features that are important to her, because she cares that the student is content with their group members and their project.

2b. Imaginary User Two

Ron Zacharski is a professor at the University of Mary Washington. Dr. Zacharski teaches many computer science courses related to data science and deep learning. Some of the courses that Dr. Zacharski teaches or has taught include: Data 101 - Intro to Data Science, CPSC 370 – Deep Learning, CPSC 405 – Operating Systems, CPSC 419 – Data Mining, CPSC 110h – Honors Intro to Computer Science, and CPSC 350 – Application of Databases. In his classes, Dr.

Zacharski often uses groups - which he refers to as teams - and utilizes an experience point learning system (XP). Because he uses an XP system, Dr. Zacharski wants each team to be fair. If teams are not fair, then teams with higher skilled members can easily gain more XP than teams with lesser skilled members. Thus, teams with more skilled members will receive higher grades than those with less skilled members. Having uneven teams will not be fair because and therefore he likes to create the teams by size and skill level. For example, if there are five teams and thirty students in the class, then six students will be in each group. However, if there were and uneven number of students in the class, then one or two groups may need to contain seven members. Through this method he tries to obtain the most equal number of members in each team that he can by organizing the teams to be as equal as possible. His second criteria would be to have at least one person in each team that has skills and experience from previous data science or deep learning courses taught at UMW. For this criterion, it does not matter if the previous courses were taught by him or different professor. Even though UMW's demographics has a higher female to male ratio, the computer science department is not heavily female. Therefore, he also has a criterion where at least two females need to be in a group together. If one female is left out and there is a group with two females, then the female can switch with a male in the other team. However, if there is only one female in the class, then she will have to be in a group with all males. Overall, creating groups by team size, skill level/experience level, and gender are all the features that are important to Dr. Zacharski.

2c. Imaginary User Three

Gusty Cooper is an adjunct instructor at UMW. Professor Cooper often utilizes group projects in his courses. He teaches courses such as: CPSC 240 – Object-Oriented Analysis & Design and CPSC 305 – Computer Systems and Architecture. In Professor Cooper's courses he usually allows for people to have groups of three people. Although, depending on the size of the class, there may be a group or groups of four. Professor Cooper is a nice professor and like Dr. Anewalt he wants students to be able to work with who they want. Furthermore, like Dr. Zacharski, he wants groups to be fair in terms of skill level. Although Professor Cooper wants students to work with who they want, he would rather have students work with other students that they do not already know in the class. This is because he wants the learning environment to be close to one like in a real work environment where you are working with strangers. However, Professor Cooper does want all the students to get along and therefore he usually creates groups by personality. To do this, he would have his students take a personality test which will assign them a color for their personality. Each color would represent a different personality type (orange, green, blue, gold). Creating groups by personality is the main feature Professor Cooper would want and this is because he cares about having the students get along well with one another.

3. User Model

3a. Overview of Software's Requirements, Functions, and Goals

This software has a list of requirements, functions, and goals that it must have. The first requirement is that it must be able to read an input CSV file created from data gathered through a Google Forms survey. This form will be passed into the software as an argument. The software then must be able to write an output CSV file with sorted groups numbered and containing the names of the students in each group. Furthermore, it must be able to write an output CSV file with multiple sets of groups for a single course section, which the user can choose to finalize or make changes to. The software must be able to write the output CSV file into the local storage of the user's machine, into a directory of their choosing and will run within the same directory as the CSV file designated by the user. The next requirement is that the software must prompt the user for specific parameters for group making use including but not limited to: group size, number of groups, randomization, student "blacklists", and group generation by similarities or differences. One function that is not a requirement is that it will utilize through the command line and display menu options. The end goal for this is that the software will have a graphical user interface, rather just a command line. The software will allow the generated sets of groups to be shown to user before being written to an output CSV file. Along with this function, the user will be allowed to make changes to the groups after the outputs is shown to the user, but before the output is written to an output CSV file. The software must display an error message when not given the correct information. For each metric the user uses to create groups, the software will ask if the groups should be formed by using similar or different values of that metric. If there aren't enough people with similar or different values to make groups, then the student will be placed within the next group that he/she best fits, or his/her group will be decided by the professor.

3b. Selected Feature and Focus Group Feedback

The one feature that I selected for my focus group to review and that my software will support are the options for choosing the criteria when creating the groups. I chose this feature because it is one that I can get the most feedback from compared to the other features. For example, the feature for opening the CSV file and the feature for getting the group size would all result in very similar feedback from each member of the focus group. Therefore, receiving the information about this feature is important. This is because I can know what other people assume, so that I can make adequate software. Furthermore, their feedback may influence my opinions on how I believe that the software should be made.

My focus group consists of four people. None of the people in my focus group are computer science majors and have never taken a computer science course. The first person in my focus group is my younger brother Charlie. The second person in my focus group is my older sister Janet. The third person in my focus group is my friend Shane. The fourth person in my focus group is my friend Angel. All the people in my focus group are around the same age and I picked two boys and two girls to be unbiased.

3b.1. How would you interact with the software?

1) **Charlie's feedback** – Charlie mentioned that there should be buttons and check boxes or drop-down menus that the user can click on. He mentioned that it should be simple and have similar features to any generic website, such as a form that people fill out online.
2) **Janet's feedback -** Janet specified that the software should be organized where it is easy to follow and know what to do. She also stated that the user can click on the file they want to use and press different buttons to get output. She also mentioned that you should be able to save the information.
3) **Shane's feedback** – Shane mentioned that the user should be able to click on what they want, and the software should be able to output the results that the user was looking for.
4) **Angel's feedback** – Angel stated that the user should be able to press different buttons within the software and get different results.

3b.2. What should you be able to do with the software?

1) **Charlie's feedback** – Charlie stated that the if he was the user he would want to have the software layout to be simple so that he could easily interact with the features.

2) **Janet's feedback** – Janet mentioned that if she was the user, she should be able to click on the different objects in the software that she wants to interact with and get the results that she wants.
3) **Shane's feedback** – Shane replied to this question by mentioning that the user should be able to perform the actions of the software and receive expected output.
4) **Angel's feedback** – Angel answered this question by stating that when the user presses each button they will receive a different result for each.

3b.3. What should you not be able to do? What is important to you?

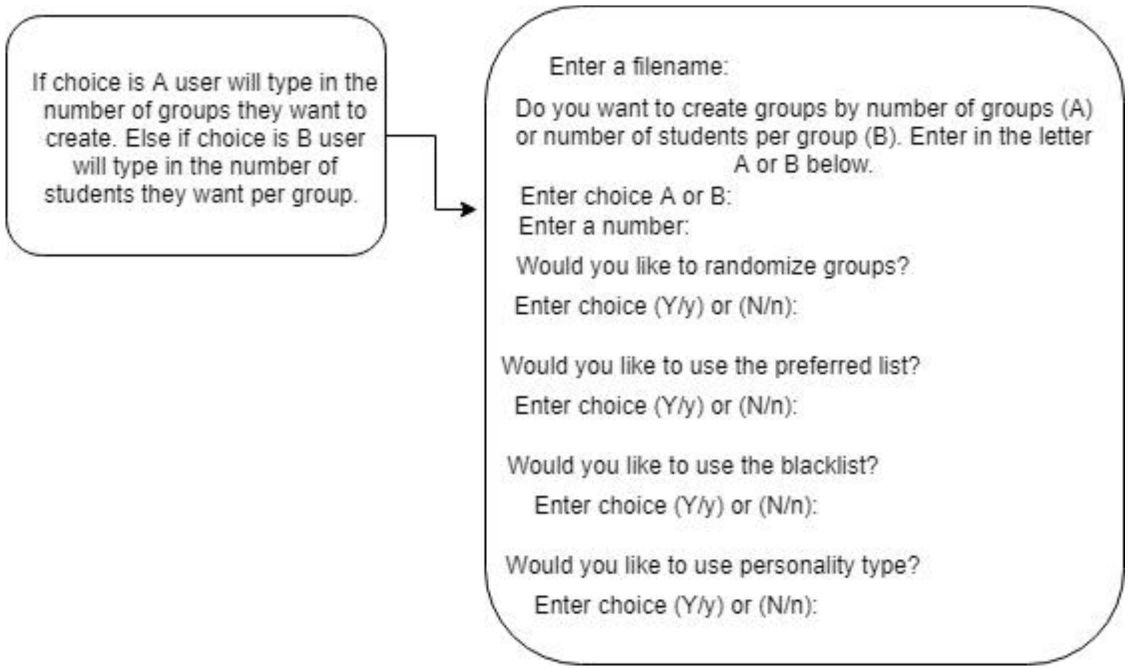
1) **Charlie's feedback** – Charlie mentioned that the user should not be able to click on anything that they are not supposed to click on causing an error. He also mentioned that the user should not be able to enter in a value for the group size if not a number. Charlie stated that it was important for him that the software was simple and easy to use.
2) **Janet's feedback** – Janet mentioned that the software should not crash and that it should work. She stated that it would be important for her as the user that the software works and is easy to follow.
3) **Shane's feedback** – Shane specified that the software should not close unexpectedly when using the software. He told me that it was important for him for the software to be neat and organized.
4) **Angel's feedback** – Angel stated that she was unsure of what it should not be able to do. However, she did mention that it was important for her as the user to have "colorful" and "pretty" buttons.

3c. Summary of user model

Overall, the focus group gave good feedback. Although, much of the feedback overlapped with one another. Even though most of the feedback overlapped, there was still enough feedback that didn't. When asked, "How should you (the user) be able to interact with the software?", many responded by mentioning that there should be buttons/checkboxes/drop-downs that the user should click on and be able to press. This is most likely because this is the first thing that people think of when using something such as a form online. Another commonality mentioned is that the software should be simple, organized, and easy to use. I believe that this is a common response because nobody wants to use software that is difficult and confusing to the user. The last thing mentioned when asked this question is that the user should get the expected results when selecting each different option. This response is understandable, because as both the developer and the user, you want the software to preform how it is supposed to and is intended. When asked, “What should you be able to do with the software?”, I received similar results to the first question. This is that the software should be simple and easy to interact with, be organized and easy to find what you are looking for and behave the way that the user expects. When asked the final question, “What should the user not be able to do?”, I received mixed answers. These answers included things such as the user should not be able to type in any value that they are not allowed, such as a typing a letter in the number field. Furthermore, the software should not crash on the user or behave unexpectedly.

4. Prototype Image/s

4a. Command Line



If choice is A user will type in the number of groups they want to create. Else if choice is B user will type in the number of students they want per group.

```
Enter a filename:
Do you want to create groups by number of groups (A)
or number of students per group (B). Enter in the letter
A or B below.
Enter choice A or B:
Enter a number:
Would you like to randomize groups?
Enter choice (Y/y) or (N/n):

Would you like to use the preferred list?
Enter choice (Y/y) or (N/n):

Would you like to use the blacklist?
Enter choice (Y/y) or (N/n):

Would you like to use personality type?
Enter choice (Y/y) or (N/n):
```

Figure 4.1 – This figure displays a command line that the user can interact with. Each line will be prompted to the user one at a time. When the user types in a valid response it will prompt the user with the next line. If the response is not valid - meaning that it is not one of the choices available – the software will repeat the prompt until a valid response is given.

4b. GUI

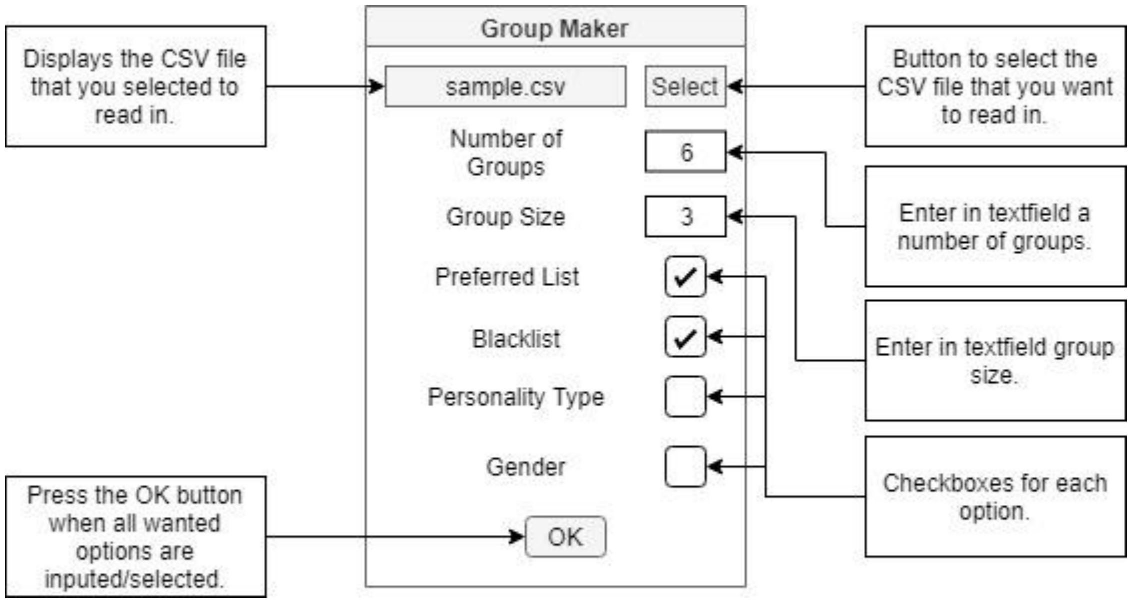

Group Maker
Displays the CSV file that you selected to read in.
sample.csv
Select
Button to select the CSV file that you want to read in.
Number of Groups
6
Group Size
3
Enter in textfield a number of groups.
Preferred List
Blacklist
Enter in textfield group size.
Personality Type
Gender
Checkboxes for each option.
Press the OK button when all wanted options are inputed/selected.
OK

Figure 4.2 – This figure displays the user interface when the project is first opened. The software allows the user to select a CSV file to read in as input and allows the user to select the criteria they want to use to create the groups.

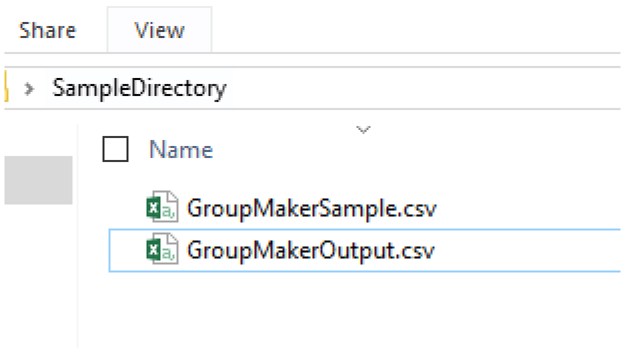


Figure 4.3 – This figure displays the directory being opened when the user clicks on the select button. The user can choose which CSV file they want to read in as input. When the file is selected the software will go back to the main user interface and display the name of the file within the field.

5. Focus Group Feedback

5a. Comments on Prototype Images

1) **Charlie's feedback** – When reviewing the images from section four of the document, Charlie mentioned that what would happen if of the fields were empty, such as blacklist or preferred list. If this were to happen what would the software do to solve the issue. He also recommended that I should remove the area for inputting in the number of groups. This is because there can be a situation where there are more classmates than the number of groups wanted, and it doesn't make sense to specify both number of groups and group size. Instead, he stated that the number of groups should just be decided by the class size divided by the group size.
2) **Janet's feedback** – After reviewing the images from section four, Janet mentioned that it was pretty much as she expected. The only thing she suggested to add was a button to save the information before finalizing the options.
3) **Shane's feedback** – After looking over the images, Shane suggested that I should make it more explicit to show where to input the CSV file. He recommended that I change the select button to say "open". This is because he found it a bit confusing on what it should do when the button is pressed. He also suggested that each criterion should have some type of help button or display a help menu in order to give a more detailed description when using the software. For example, people will not know what the criteria is for personality type.
4) **Angel's feedback** – When looking at the images, Angel did not have much feedback and thought that the images looked good and was more than what she expected.

5b. Reflection on Feedback

5b.1. - 5b.2. Commonalities & Discrepancies

After the focus group reviewed the images of the user interface there were not many commonalties like the feedback in section three. However, there were a lot of discrepancies when it came to what each button is supposed to do. There seemed to be a lot of confusion of what each criterion was supposed to do, even with the explanation on the sides. Some words brought confusion, such as using "select" instead of "open" to input the CSV file. Other feedback suggested that things were missing such as a help button and a save button.

Furthermore, one response mentioned the confusion of including both number of groups and people per group. This feedback made sense, because the user would have to know specifically how many students were in the class and exactly how many groups they want to create with the given class size. Also, it would not work if the user specified for example that they want five groups and three students per group, but there were either more than or less than fifteen students in the class. Therefore, I made the following changes to my user interface below in section six.

6. Changes

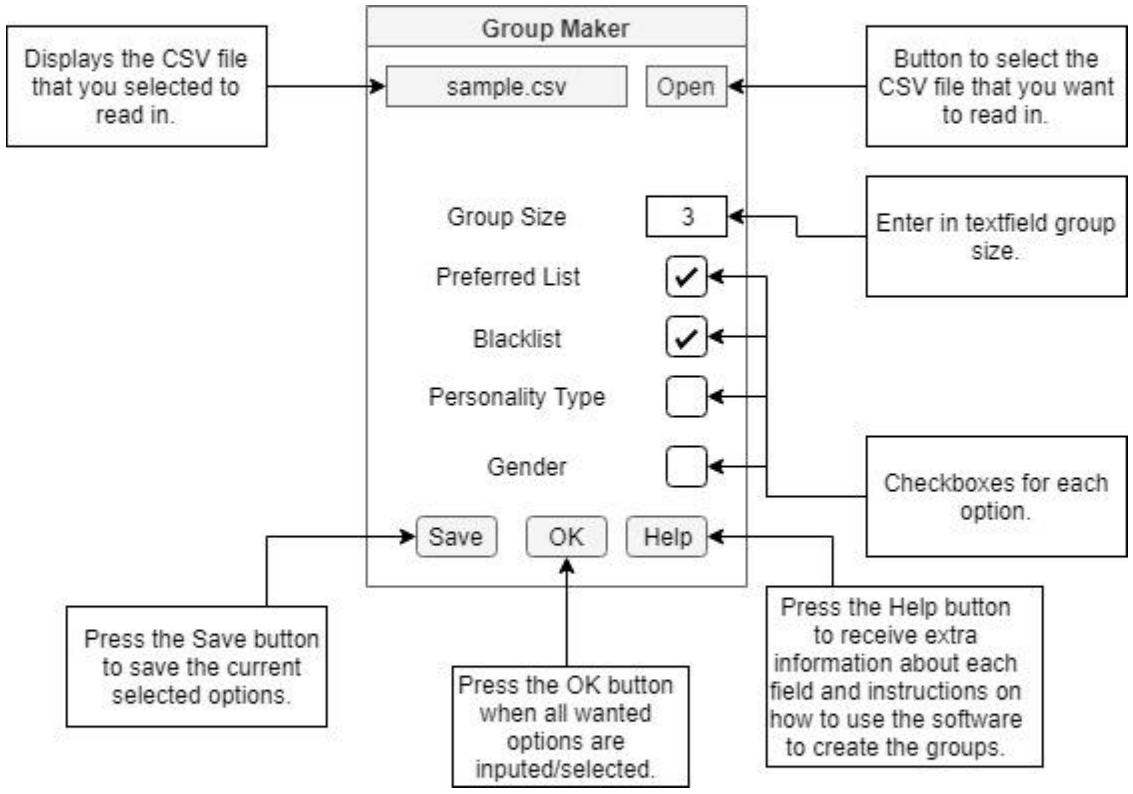

Group Maker
Displays the CSV file that you selected to read in.
sample.csv
Open
Button to select the CSV file that you want to read in.
Group Size
3
Enter in textfield group size.
Preferred List
Blacklist
Personality Type
Gender
Checkboxes for each option.
Save
OK
Help
Press the Save button to save the current selected options.
Press the OK button when all wanted options are inputed/selected.
Press the Help button to receive extra information about each field and instructions on how to use the software to create the groups.

Figure 6.1 – This figure displays the changes made to the graphical user interface. The image of the software now displays a save button to save changes before finalizing, and a help button so that the user can have a better description of what each criterion does. Additionally, the button for "select" is now changed to "open". Furthermore, the field for creating the number of groups is removed so that groups will be created by the number of classmates divided by the specified group size.