**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Software Requirements Specification**

for

**Group Maker**

**Version 1.0 approved**

**Prepared by:**

**David Zamojda**

**Aaron Bloomfield**

**Clay Sweetser**

**CPSC430**

**01/29/19**

**Table of Contents**

[**1. Introduction**](#_hekzkvyd66x8) **2**

[1.1. Purpose](#_lr8zadz9c82j) 2

[1.2. Scope](#_za7w8bpunws) 2

[1.3. References](#_yqendzz3v5ko) 2

[1.4. Document Overview](#_ijh3zxm3csns) 2

[**2. Project Description**](#_ne7clqblqnt4) **3**

[2.1. System Overview](#_tof9dc1r5qyh) 3

[2.2. Client Characteristics](#_3cqwbgkpevq9) 3

[2.3. User Characteristics](#_d2cr3gh8xg7c) 3

[**3. Requirements**](#_z8w2ta9dsxyu) **4**

[3.1. Functional Requirements](#_jn0pu7i9jvsi) 4

[Overview](#_g9wa6c2850k9) 4

[Reading](#_w3f1z99oclwj) 4

[Writing](#_10vxp6h9mawx) 4

[Options](#_z6h4e1t7gs6k) 5

[User Interface](#_chv94hkxn7g7) 5

[Function of the Software](#_qyhc6pz21n4n) 6

[3.2. Non-Functional Requirements](#_y35hrop6j51f) 7

[Overview](#_9g79wnjbcb72) 7

[**4. Non-requirements**](#_1o40xfo9w7vs) **8**

[**5. Assumptions**](#_9ckmjivh3x0w) **8**

[**6. Appendices**](#_xpz8vhz6y0b7) **8**

[6.1. Glossary of Terms](#_92krkak3utfl) 8

[6.2. Author information](#_pd9uzrm1ar1) 8

[6.3. Additional Documents](#_94rd6khzpe60) 10

[Example User CSV From Google Survey](#_gw5ksbb6kvg) 10

[Example User Modified CSV](#_dzabtcntkhz9) 10

[Example Group Output](#_9eeb6gym400x) 10

## 1. Introduction

### 1.1. Purpose

The purpose of this document is to serve as a comprehensive overview and description of the various features the software should have. This reference is intended for review by both the developers of the software, and the client/clients requesting the software.

### 1.2. Scope

The requested software's purpose is to form individual groups from lists of individuals, given a set of metrics describing those individuals and a description of how the groups should be formed from those characteristics.

### 1.3. References

**1.3.1** - The Project Description Paper

### 1.4. Document Overview

The remainder of this document split into the following sections:

* Section 2, "Project Description", in which the software's characteristics, users, clients, and functions are discussed.
* Section 3, "Requirements", in which both the functional and non-functional requirements of the software are listed and described.
* Section 4, "Non-Requirements", in which a list of features that will *not* be included in the software is given.
* Section 5, "Assumptions", in which a set of assumptions about the software, including how it will be used, who will be using it, etc. are given and described.

## 2. Project Description

### 2.1. System Overview

The software developed will be an input-based group-making software. Its purpose will be to take data from a CSV file, such as one generated from a Google Forms survey, then create groups of students based off of certain metrics chosen by the user.   
These groups may then either be sorted by an additional set of metrics, or sorted randomly. Examples of metrics include, but are not limited to, whether students commute or live on campus, the students' personality traits, and the students' progress in their degree program.   
After groups are made, the software will write a CSV file to a directory of the user’s choosing. This CSV file will the groups generated by the software.

### 2.2. Client Characteristics

The client is Professor Karen Anewalt, Ph.D. Dr. Anewalt’s request for such a software comes from managing multiple classes, each of which require groups of student to be generated at least once during the semester.  
This software will enable quick generation of such groups.  
In addition, this software will also help reduce any unconscious bias involved in the group generation process through use of randomization.

### 2.3. User Characteristics

The intended users of this software will be the University of Mary Washington’s computer science faculty. Students will not have access to this software, only professors and faculty.

## 3. Requirements

### 3.1. Functional Requirements

#### Overview

* Functional requirements for this software include the ability to read an input CSV file containing student information and the ability to write a CSV file containing all groups sorted. When a CSV file is read by the software, it will display a set of options for the user to select from on how they would like the groups to be generated. These options will be displayed through the user’s terminal interface. Once an option is selected, the software will generate a group or multiple groups based off of the criteria selected by the user.

#### Reading

**3.1.1. -** The software must be able to read an input CSV file created from data gathered through a Google Forms survey, passed in as an argument.

* As a user, I want to be able to pass a CSV file created from a Google Forms survey, so that the software will have all students that I want to put into groups.

**3.1.2. -** The user must run software within the same directory as the CSV file.

* As a user, I want the CSV file in the same directory as the software so that the software will have access to the CSV file.

#### Writing

**3.1.3. -** The software must be able to write an output CSV file with sorted groups numbered and containing the names of the students in each group.

* As a user, I want the software to write a CSV file that contains all groups generated by a single run of the software, with each student's name and each group sorted by numbers.

**3.1.4. -** The software must be able to write an output CSV file with multiple sets of groups.

* As a user, I want the software to write a CSV file that contains multiple possible sets of groups instead of just one group set.

**3.1.5. -** 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.

* As a user, I want the software to export the CSV file into the same directory as where the original file was, or the directory of my choosing for my convenience.

#### Options

**3.1.6. -** The software must prompt the user for specific parameters for group making use including but not limited to: group size, number of group sets, randomization, student “blacklists”, group generation by similarities or differences, and whether students should be included in more than one group.

* As a user, I want to be able to give the software the necessary information when prompted in order to get the best results possible.

#### User Interface

**3.1.7. -** The software must be utilized through the command line.

* As a user, I want to be able to use the software through the command line as it is straightforward and easy to use.

**3.1.8. -** The software must display menu options through the command line.

* As a user, I want the software to display the prompts for my input through the command line.

**3.1.9. -** The software will allow the generated sets of groups to be shown to user before being written to an output CSV file.

* As a user, I want to be able to look at the generated sets of groups before they are written to make sure that everything looks correct.

**3.1.10. -** The software will allow the user to make changes to the groups after being shown to the user, but before being written to an output CSV file.

* As a user, I want to be able to change the groups if I spot a problem with them before the groups are finalized.

#### Function of the Software

**3.1.11. -** The software must be able to sort groups based off of a selected value.

* As a user, I want to be able to to sort groups in ascending or descending order, based upon various attributes of the group.

**3.1.12. -** The software must display an error message when not given the correct information.

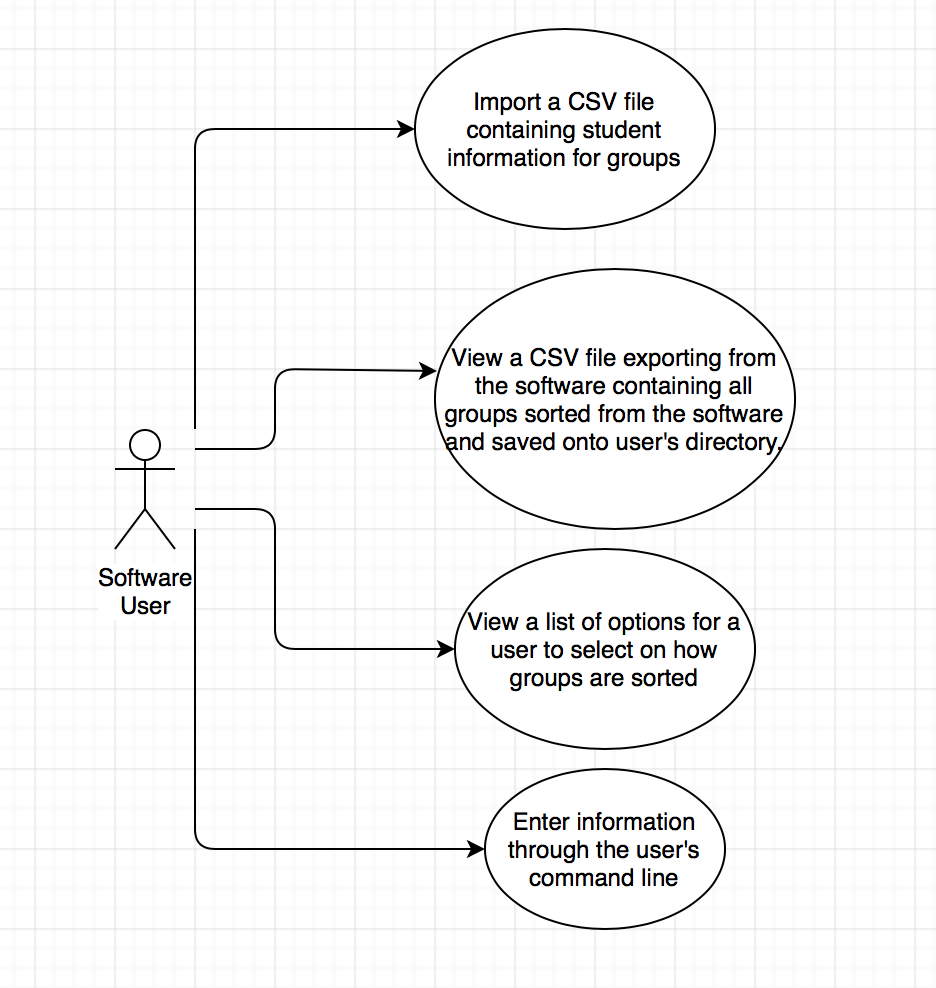
* As a user, I want to receive a clear, informative error message when I enter invalid input into the software, or when something goes wrong with the software.

**3.1.13. -** The software will display the following questions in this order: “How many groups will be made?”, “How many sets of groups will be made?”. If there will be more than one set of groups, software will ask “Will there be repeated members within groups?”.

* As a user, I want to specify the size and/or number of groups to create, as well as whether a single individual can be part of more than one group.

**3.1.14. -** 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.

* As a user, I want to be able to create groups formed from a metric with similar or different values.



### 3.2. Non-Functional Requirements

#### Overview

* The non-functional requirements for the software are limited. The budget allocated for the software is $0.00. The security of the software is limited to the user’s discretion since only the University of Mary Washington faculty will have access to the software. The performance of the software should be able to perform all sorting tasks within 5 seconds, 95% of the time.

**3.2.1. -** Budget allocated for this project is $0.00 .

* As a stakeholder, the budget set for this software project is $0.00 because that is what is left in the budget for the semester.

**3.2.2. -** The software must have a runtime of less than five seconds 95% of the time after all information has been collected from the user.

* As a user, I want the software to have a runtime of less than 5 seconds, as I want groups to be made almost immediately.

## 4. Non-requirements

**4.1. -** The software will not store any data outside of the user’s own computer.

**4.2. -** The software will not require the user to sign in as a faculty member.

**4.3.** - The software will not provide functionality for creating or modifying a set of input data.

## 5. Assumptions

**5.1 -** There will only be one instance of the software running at one time..

**5.2 -** The user will have some competency running programs through the command line.

**5.3 -** The user will be using a CSV file that has been made from a Google Forms survey.

**5.4 -** The user will be UMW faculty.

## 6. Appendices

### 6.1. Glossary of Terms

**6.1.1** - CSV: Data following the Comma-Separated-Value format.  
Information on the CSV data format can be found at <https://en.wikipedia.org/wiki/Comma-separated_values> .

**6.1.2** - UMW: University of Mary Washington

### 6.2. Author information

Clay Sweetser

* Non-Requirements
* Glossary of Terms
* One-third of the Requirements/User Stories
* Introduction
* References
* Overview
* Final Formatting and Grammar Revision

David Zamojda

* Non-Requirements
* Assumptions
* Created the additional documents
* Title page and table of contents
* One-third of the requirements

Aaron Bloomfield

* Project Description
* Functional Requirements Overview
* One-third of the Requirements and user stories
* Part of Non-Functional Requirements
* Use-Case Diagram

### 6.3. Additional Documents

#### 6.3.1 - Example User CSV From Google Survey

#### 6.3.2 - Example User Modified CSV

#### 6.3.3 - Example Group Output