# Software Requirements Specification

Project: Group Please

Cody Hubbard - 004843389

# 1. Introduction

This section gives an explanation of everything that is in this requirements document. It also serves to provide a “big picture” of the application.

## 1.1 Purpose

The purpose of this document is to provide a detailed description of the requirements for the “Group Please” (GrPl) software. It provides a description of the features and use cases of the software as well as a UI mockup and overview of the technologies used. This document is designed to be shared with the client as a project proposal to be approved, or changed as needed. It also acts as a base reference for the development team during their creation of the software.

## 1.2 Scope

The “Group Please” software is a web application that helps instructors group students for projects based on a set of preferences supplied by both the instructors as well as the students; Including but not limited to: group size, number of projects, number of teams per project, student project preferences, and student teammate preferences.

The primary goal is to group students so that as many of their preferences as possible can be satisfied, resulting in “happy” students while still satisfying the group/project rules set by the instructor.

Lastly, the software should be accessible through the courses website where the Instructor can create a new group project where they input a number of attributes. In this vein, after a project/assignment has been set up students can login to their unique account and submit preferences. The final result returned by application is a set of student teams and the project matched to each team.

## 1.5 Overview

The rest of this document contains four main sections. Section two provides a more detailed overview of the software functionality, software objectives,user roles, outside systems interactions and dependencies, and software design and implementation constraints. Section three contains interface requirements, UI mockups, and the interaction of all input into and all output out of the GrPl software. Section four countain functional requirements, and section five contains the non-functional requirements.

# 2. Overall Description

## 2.1 Product Perspective

Group Please will consist of one main part, a web app that will host instructor made assignments where students will be able to submit group and project preferences.

The web app will be entirely internet hosted, meaning it will just act like a website and all users will need an internet connection to have any meaningful interaction with the software.

There are several types of information that will need to be stored for GrPl to function. Unique user accounts will need to be created and stored. Instructor account linked assignments will need to be created and stored. Student preferences will need to be stored. Final group assignments will need to be stored. For all of this data storage we will use a database. The GrPl software will rely heavily on this database. It will create, modify, and store data very often. All of this database communication will happen over the internet.

## 2.2 Product Functions

The web application will need to have a final output of optimal group and project matches that is accessible and modifiable by the instructor. This is the primary function of the software.

Since this is the final output the software will need several other functions to lead up to the generation of this output. Firstly unique user accounts will need to be able to be created and stored. Instructor and students will have different account types. Instructors will be able to create assignments and input assignment characteristics such as project names, groups per project, and optimal group size, and submission deadlines. Once an assignment is created student who are enrolled in the course will be able to submit their project and group preferences.

The application will provide functionality for instructors to manage assignment information at any time. They will be able to modify any and all characteristics as needed. Student will be able to modify their account information and resubmit their preferences at any time before the assignment deadline.

## 2.3 User Characteristics

As mentioned before there will be two primary types of users, instructors and students. Each will require a different types of functionality so they have seperate requirements.

The instructors will have an assignment where multiple student groups need to be formed. The instructor will need to have multiple projects/teams in that students can be assigned to. They will also need to have a group size in mind as well as a deadline for when the group matches must be formed by. The instructor will have to submit all of this information when creating the class assignment.

The students will have to submit preferences to assignments created by their instructor. They will be able to select their preferences from a list projects submitted during the assignment creation by the instructor. If allowed by the instructor group member preferences will be able to be submitted as well from a list of other students enrolled in the class.

Boh types of users will use the same web application, but the instructors will need to have more controls to manages a plethora of assignment and class information while students will only need to manage wither project submissions.

## 2.3 Operating Environment

The web application shall be supported by the big modern browsers, Chrome, Firefox, Safari, Edge, etc. A mobile version of the site is not needed.

Since it is a web app there must be a constant internet connection for the software to function. The accessing and submission to the database for all tasks needs an internet connection. The final computation of optimal group/project matches will be done server side and also require an internet connection.

The web app will support normal keyboard and mouse input.

## 2.4 Constraints

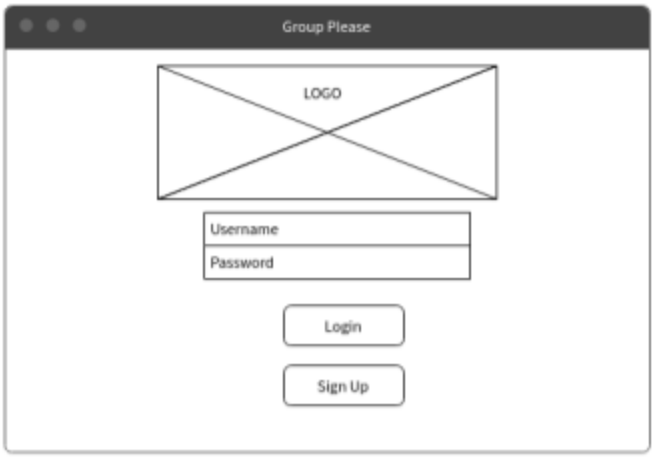
The largest constraint is the need of a constant internet connection for the software to function correctly. The database accesses and submissions will need an internet connection to be made.

The web app will have a major constraint in the capacity of the database. Another constraint is account security. Since both types of users need unique accounts we will need a way to store account information securely and ensure user connections to our web app are secure. This may slow down connection or login times and limit account creation.

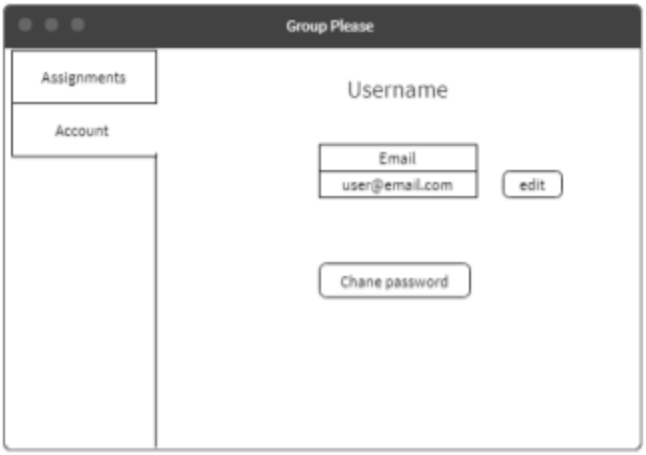
# 3. External Interface Requirements

## 3.1 User interfaces

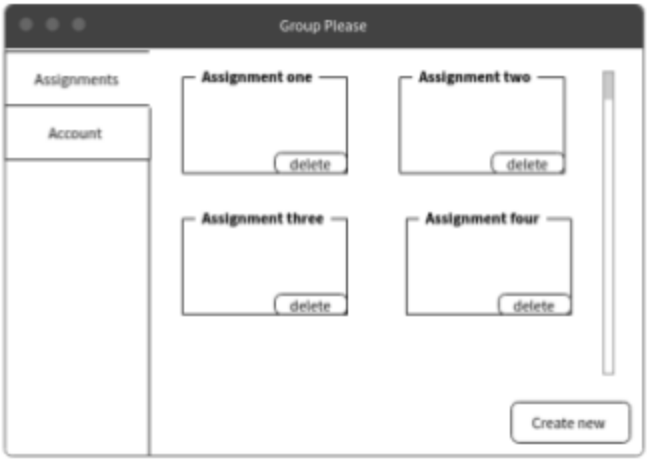
Group Please will have a basic website UI with several pages. A general homepage that displays available class assignments to submit preferences for. A variation of this page for instructors where new assignments can be created. A assignment creation page. A submission page for student preferences. An account creation page, and a account management page.



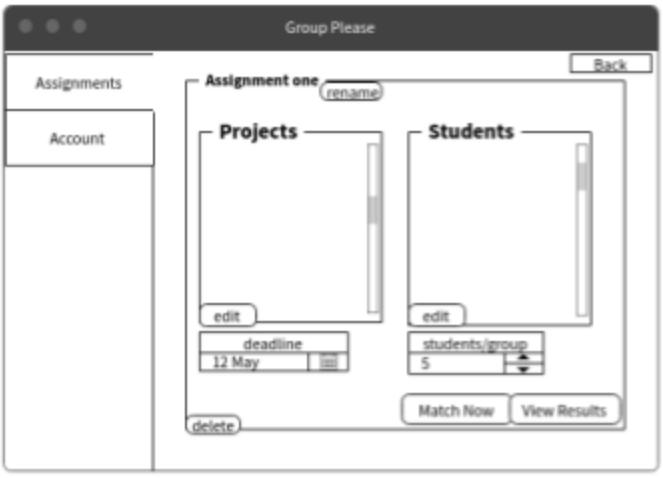

Login Page



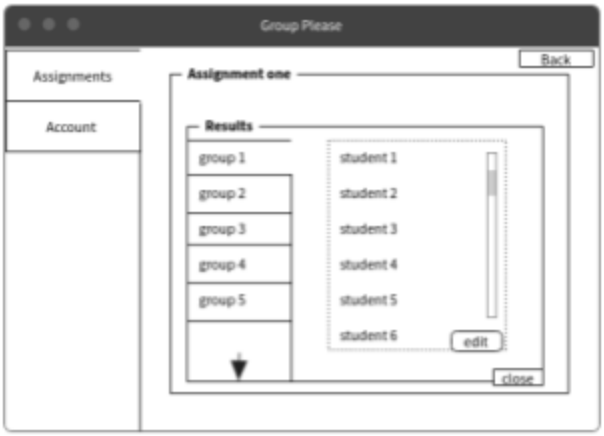

Account management page



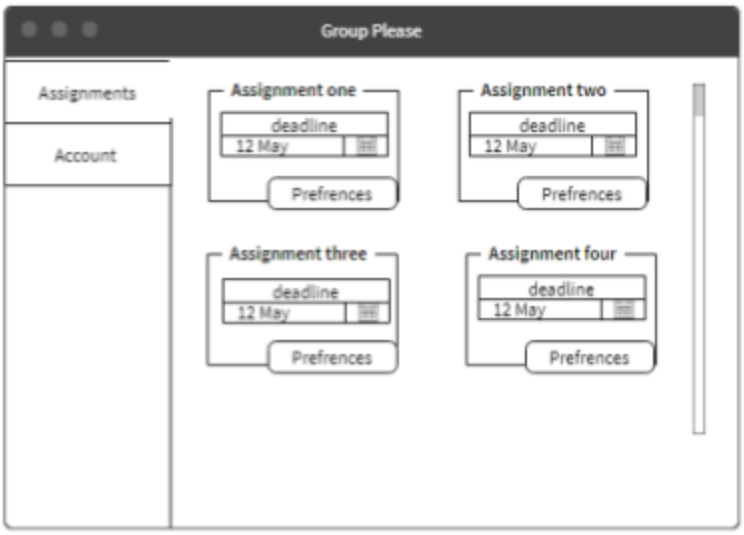

Assignment View - Instructor



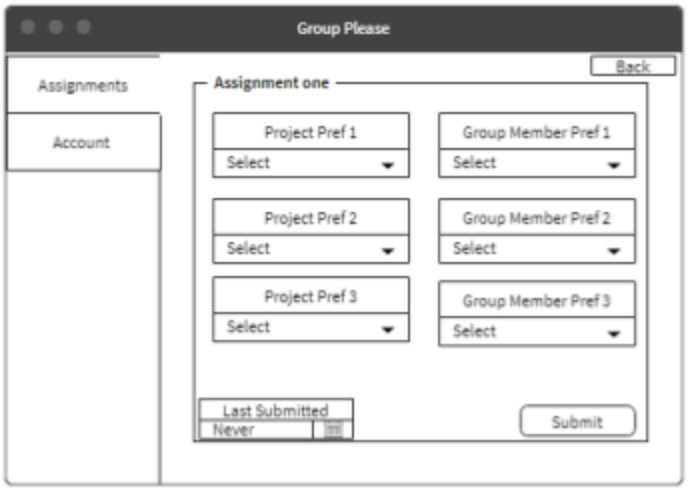

Individual Assignment - Instructor




Individual Assignment Group Matching Results




Assignment View - Student




Student Preference Submission Page

The first time a user comes to the website they should be greeted with the login page. They will be able to login with an existing account or create a new account. All users will be shows the same “Account” page where they can manage their account information.

Once a user has logged in they will see the main page, which is different depending on the user level. Students will see the student version where they can view a list of assignments belonging to the classes they are enrolled in. Once an individual assignment “preferences” button is clicked the “Student Preference Submission Page” will be shows where students can select their preferences

from a drop down list of projects/students added to the assignment by the instructor. Once all fields are filled out a student can submit their preferences by clicking the “submit” button. If the form is attempted to be submitted without all fields filled out a dialogue reminding students to fill out all the fields will be created. The last submitted date field will be updated accordingly once a successful submission has been made. Students will be able to submit with no limits until the deadline.

The instructors main page has more assignment control features. They have the ability to create new assignments and delete existing ones. Once an Instructor clicks on a specific assignment they will be able to see and edit the assignments properties. An instructor can click the “match now” button to compute the optimal student/project where they will be displayed after a short delay. Previously calculated machines can be shown with the “View Results” button. Lastly next to each students name in the instructors specific assignment view there will be an indicator next to them indicating if they have submitted preferences or not, furthermore a students indicator can be clicked to show the preferences they selected.

### 3.2 Software interfaces

The software will communicate extensively with its own database for most functionality. The communication will need to facilitate both reading and modification of data pertaining to instructor assignments and student preferences. Furthermore user account login and creation can be streamlined by linking user accounts within GrPI to users respective institution accounts.

# 4. Functional Requirements

## 4.1 All Users

### 4.1.1 Logging in

1. Users shall be able to access the website from any typical internet connection.
2. Users shall be able to login to an existing account or create a new one.
3. Users shall be logged in, given they have input a correct username and password, when the “login” button is clicked.
4. Users shall be told when an unsuccessful login attempt was made
5. Users should be taken to an account creation page when the “Sign Up” button is clicked.

6. When the user has logged the “assignments” left side tab shall be selected by default.

### 4.1.2 Account Management

Dependendencies:

A user has logged into their Group Please account.

Context:

A left side tab is one of the titled left side UI elements as seen on the UI mockups.

1. Users shall be taken to the account management page when the left-side “Account” tab is clicked.
2. On the account tab a user's username shall be displayed.
3. On the account tab a users email address shall be displayed
4. On the account tab a user shall be able to change their email address by clicking the “edit” button and then inputting a new email into a newly created dialogue box.
5. On the account tab their password in NOT displayed.
6. On the account tab a user can change their password by clicking the “change password” button.
7. Once the change password button is clicked a new dialogue box will be created asking the user to input a new valid password.
8. Valid passwords are password which contain at least one capital letter, once special character, and are at least 7 characters long.
9. Users can leave the Account page by either clicking the back button or clicking on a different left side tab.

## 4.2 Student Users

Dependendencies:

A user has logged into their Group Please account and their account standing is student.

Context:

A left side tab is one of the titled left side UI elements as seen on the UI mockups.

1. On the assignments tab a panel display of all Instructor created assignments shall be shown (see UI mockup)

2. Each assignment panel shall have a static deadline displayed as well as a preferences button.
3. Given the number of assignments exceeds the available screen space a user can scroll through them using a typical scroll bar.
4. When the preferences button is clicked the corresponding assignments panel shall expand to take up the full window area.
5. The expanded assignment panel shall expand to show preference selection tools as shown in the UI mockup.
6. The Selections present in the respective prerence's drop down list shall be taken from data submitted by the Instructor during the creation of the assignment.
7. When the "submit" button is clicked the website will check to see if all project preference fields are filled out, if not a dialogue will be created reminding the user to fill out all fields.
8. A "no-preference" selection will be available for all preference fields.
9. Once a valid submission has been made the "last submitted" reminder shall be updated.
10. A user can leave an assignment's specific expanded panel by clicking a left side tab or by clicking the "back" button.
11. (not in mock up) A student shall be able to join classes created by instructors by searching for said class, or following a unique link given when the instructor creates a class.

## 4.3 Instructor Users

Dependendencies:

A user has logged into their Group Please account and their account standing is instructor.

Context:

A left side tab is one of the titled left side UI elements as seen on the UI mockups.

1. On the assignments tab a panel display of all user created assignments shall be shown (see UI mockup).
2. Given the number of assignments exceeds the available screen space a user can scroll through them using a typical scroll bar.
3. Each assignment panel shall have a small delete button will be displayed in the bottom right corner.
4. If the user clicks the delete button two sequential delete verification dialogues shall be created.

    a. The first delete verification will ask the user to correctly type the assignments name.
    b. Once the first dialogues task has been correctly fulfilled a “are you sure” yes/no dialog will be created.
    c. If yes is selected the assignment and all of its data will be unrecoverably deleted. If no is selected the view will return to the default assignment view.
5. When the assignments tab’s “create new” button is clicked a new dialogue shall be created asking for a deadline and an account unique assignment name. The submit button on this new dialogue will not work until these two fields have been filled.
6. Once a new or existing assignment has been clicked it shall expand to show the assignment’s complete information and its control interface.
    a. Assignment information includes lists of students and projects for that class assignment, the deadline for submitting preferences, and the number of students per team.
    b. This information needs to be entered by the instructor once an assignment is created. That is, these sections are empty at first and must be filled out before a student can submit any meaningful preferences.
    c. Assignment projects can be changed by pressing the edit button.
    d. (preferred functionality) The student list for an assignment should be autofilled from a list of student enrolled in that class voluntarily.
    e. The minimum number of students per group can be set and changed at any time. The matching process will make groups with at least this many members. It has a default value of 4.
    f. The assignment deadline can be changed at any time to any date that is after the current date and time. The deadline sets the final date for students being able to submit/change preferences.
    g. When the deadline passes the software should run a student/project matching with the current student preferences and assignment attributes.
7. When an assignment’s “Match Now” button is pressed the web app will compute optimal student/project matches with the current specifications. This button shall be able to be pressed an infinite number of times.
8. The results of the last matching for and assignment shall be stored and viewable by pressing the “View Results” button for that assignment. For matching view layout see the mock up.

a. When the user is viewing the results of the software made student/project pairs they shall be editable by pressing the “edit” button when for each specific pairing. Both project and students can be changed.
b. When there is a conflict in student assignment the user will be given a warning, and the new edited pairings will not be saved until all conflicts are resolved. The user shall be made aware of this restriction.

9. (not in mock up) Instructors shall be able to create unique classes students can join. When a new class is made a unique link will be generated student can follow to join that class and it becomes findable thought the student’s class search function.

# 5. Non-Functional Requirements

## 5.1 General Use

1. When the matching algorithm is run it shall fulfill as many students top preferences as possible, making students as “happy” as possible.
2. The matching algorithm should run in less than a minute.
3. All communication must be secure, unsecure connection will prevent login.
4. All password and login data shall be securely stored.