



TEACHER'S PET  
KEEPING FRIENDS TOGETHER

# TEACHER'S PET

User's Manual

Frank Liu, Front Edge Software Pty Ltd

[www.teacherspet.net.au](http://www.teacherspet.net.au)



# Contents

1.	Introduction.....	3
	About this manual.....	3
	Learning to use the portal .....	4
	Teacher's Pet in a Nutshell .....	5
2.	Administering the School Account .....	6
	Registering the school account .....	6
	Types of user roles .....	9
	Add administrators and teachers.....	10
	Editing an existing user.....	11
	Changing the primary administrator.....	12
	Paying for subscription.....	13
	Selecting a role when logging in.....	15
3.	Step 1: Picking the Class Allocation Criteria .....	17
	Class-allocation criteria.....	18
	What if my criterion is not listed?.....	21
	What is a class-allocation task.....	21
	Cross-task data vs. task-specific data .....	22
	Creating a new task .....	23
4.	Step 2: Entering Student Data.....	24
	All students are displayed in a table .....	24
	Entering new students .....	26
	Two ways to enter students.....	26
	Manually entering a new student.....	26
	Defining options for each dropdown box .....	29
	Editing existing students .....	32
	Deleting students .....	33
	Friends.....	34



Three ways to enter friends.....	34
Manually entering friends (by teacher).....	34
Students picking their own friends.....	36
Inclusions & exclusions.....	41
Two ways to enter inclusions & exclusions.....	41
Manually entering inclusions & exclusions.....	41
Uploading student data file (CSV).....	45
Downloading student data into CSV files.....	49
Excluding a student from a custom/secondary task.....	50
Finalizing and unfinalizing data input.....	51
5. Step 3: Creating the Class Lists.....	54
Weightings of the criteria.....	55
Verifying the allocation result.....	56
Verifying the intelligence of the algorithm.....	58
Number of classes vs. number of friends.....	65
Sympathy toward friendless students.....	66
Dragging & dropping students.....	71
Saving allocation results.....	74
6. Appendixes.....	77
Sample student data.....	77



## Introduction

Teacher's Pet is a website based on an ingenious mathematical algorithm, which helps teachers to create class lists, keeping friends together in the new classes, meanwhile following many other class-allocation criteria, such as gender balance, academic balance, exclusions and inclusions, etc.

To create the class lists on Teacher's Pet, teachers go through three simple steps:

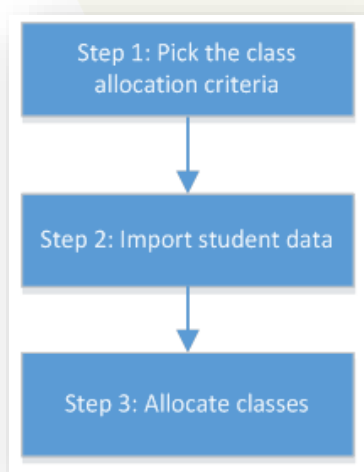


Figure 1 - Three simple steps to create class lists

Each school is required to have an account in Teacher's Pet and pay the annual subscription fee to be able to use its service. There are two types of school users: administrators and teachers. An administrator is able to add other administrators and teachers, assign teachers to grades, and deactivate users. A teacher is able to enter, edit or delete students and their data, and perform class allocations.

## About this manual

This manual introduces every aspect of the Teacher's Pet portal. Everything is introduced using screen shots, so it is easy to read and understand.

We know now busy teachers are, so this manual is no meant to be read from cover to cover, although, if it is indeed read from cover to cover, it only takes a few hours, and the user will definitely become an expert on this portal, and appreciate how simple and intuitive the portal actually is.



This manual is written to follow the sequence of the steps that a teacher needs to follow to create class lists. So a teacher reads the first section to do the first step, then the next section to do the next step, and so on. Every time the reading only needs a few minutes.

The manual is broken down into many topics, with concise and clear headings. Each topic is self-contained and complete, and only one or two pages long. The user can easily find a topic on the index, then go to read one or two pages.

Chapter “Administering the School Account” introduces the account administrative tasks, such as registering a new school account, adding new users and paying subscriptions.

Chapter “Step 1: Picking the Class Allocation Criteria” introduces the first step to create class lists – to pick the class-allocation criteria. It also introduces how to create different class lists for different events during the school year, such as camping.

Chapter “Step 2: Entering Student Data” introduces the second step to create class lists – to enter student data into the system. The data are needed by the system to do the class allocation.

Chapter “Step 3: Creating the Class Lists” introduces the third step to create class lists – to do the class allocation and control the outcome using weightings assigned to the class-allocation criteria.

Throughout this manual, a fictional school called “Frank’s Fictional School” will be used as example.

Computer-generated student names and data, such as their current classes, academic levels, desired friends will be used to demonstrate every use case. The two data files that contain such generated test data are listed in appendix “Sample student data”.

NOTE: when a figure or a chapter is referenced anywhere in this manual, the reference is actually a link. Clicking this link will jump to that referenced figure or link.

## Learning to use the portal

With features and flexibilities comes complexity. Anyone can use a lawn mower without learning, but most people needs a few hours to learn to drive a car. Teacher's Pet is like a car. It has a lot of flexibility and can do almost everything that a teacher ever wants to do when it comes to dividing students into groups following some criteria.

Because of the intuitive user interface, if the user is technology savvy, he/she may be able to use it straight-away without any prior learning, just like Josh, the first teacher to use Teacher’s Pet from end to end. However, for the majority of users, Teacher’s pet is not something they can start using without any prior knowledge or experience. Like learning to drive, they need to set aside half an hour’s time to get familiar with



the process and a few simple concepts. Otherwise the user may get stuck on places where other informed users find easy and straight-forward, thus end up spending a lot more time than the recommended learning time.

The quickest and easiest way to gain an initial understand on the portal, is to go through the YouTube videos listed on the home page. It only takes half an hour. Alternatively, if the user prefers reading, he/she can read a separate document called "Teacher's Pet in a Nutshell". It offers a high-level but thorough introduction to the whole class list creation process, and takes only ten minutes to read because it is mostly screen shots.

Then the user can start using the portal. He/she may be able to get through to the end without any problem. If stuck, he/she can go to the relevant topic in this manual for help.

After a user has gone through the whole process, he/she will realize how simple and intuitive the whole process is. An experienced user takes less than five minutes to finish the end-to-end process: from the very beginning - picking the criteria and creating the classes, to the very end - finishing the class allocation. The biggest work load is to get the students into the system. Once they are in the system, next year they don't need to be re-entered. Just change a few values, for example a student's academic level may have changed from last year. The whole process can be done in an hour's time.

## Teacher's Pet in a Nutshell

As mentioned above, there is a separate document called "Teacher's Pet in a Nutshell". It is eight pages long, but most are screen shots. It can be read in 10 minutes. It offers a complete and thorough overview of every aspect of Teacher's Pet. After reading that document, a teacher will be able to imagine how easy, comfortable and quick the class-allocation task will become by using Teacher's Pet. Most teachers may start to use the portal, and only refer to this manual when they are stuck.



# Administering the School Account

## Registering the school account

A school can only use Teacher's Pet after it registers a school account.

The person whose email and name are entered when the school is registered becomes the primary administrator of this school account. This primary administrator can later change the primary administrator to someone else.

To register a new school, on Teacher's Pet home page, click the "Register School" menu item. The "Register School" page will be shown. Fill in the school details and the CAPTCHA characters, and click "Save":

School name:

Street Address:

City:

Country:

More address details:

Suburb:

Postcode:

State/Province:

Time zone:


**Primary Administrator**

First Name:

Last Name:

Email:

Phone:



Refresh CAPTCHA  
Please enter the above text into the text box below:

Figure 2 - Registering your school

The above figure shows how the author registered the demo school called "Frank's Fictional School".

If the school is successfully registered, following message box will be shown:

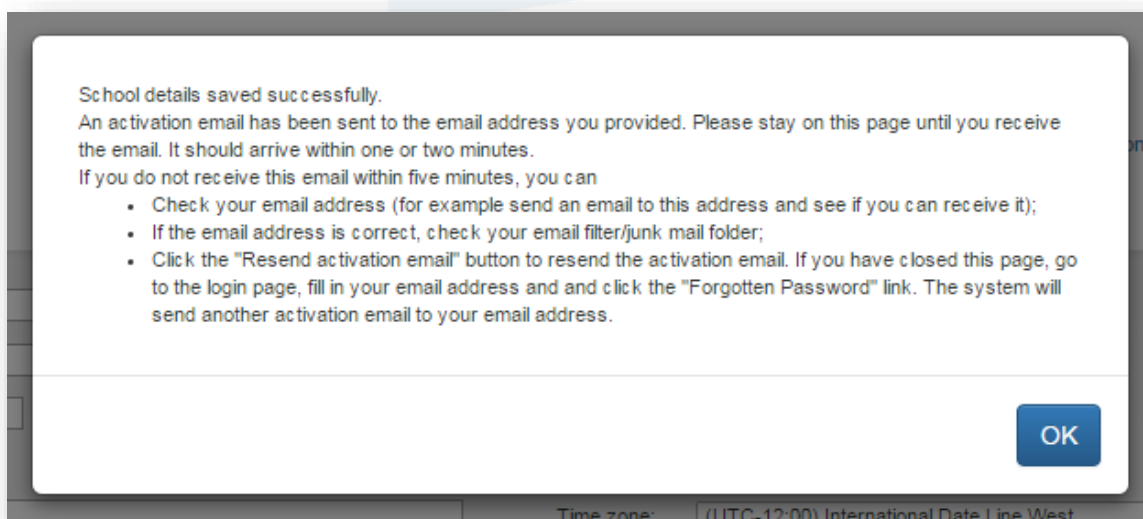


Figure 3 - Activation emails sent message box

Follow the above instruction, until the activation email is received. If none of the above works, please click the "contact us" menu item and provide us with the school name, address and email address. We will then help you to resolve this issue.

Once the activation email is received, click the link inside the email, the following page will be shown:

Figure 4 - Activate your email

Enter the desired password in both textboxes, and click "Submit". The user's account will be activated. This password will be the password which is used to login from now on.

Click the "Login" menu item on the home page to login. Following "School Admin" page will be shown:





School name:	<input type="text" value="Frank's Fictional School"/>	Active:	<input checked="" type="checkbox"/>
Street Address:	<input type="text" value="Foo Bar Street"/>	Suburb:	<input type="text" value="Melbourne"/>
City:	<input type="text"/>	Postcode:	<input type="text" value="3000"/>
Country:	<input type="text" value="Australia"/>	State/Province:	<input type="text" value="Victoria"/>
More address details:	<input type="text"/>		
School Type:	<input checked="" type="radio"/> Primary <input type="radio"/> Secondary		
<b>Primary Administrator</b>			
First Name:	<input type="text" value="Frank"/>	Last Name:	<input type="text" value="Liu"/>
Email:	<input type="text" value="silanestest3@gmail.com"/>	Phone:	<input type="text" value="0412-345-678"/>
<input type="button" value="Save"/>			

## Users

Name	Email	Email Verified	Phone	Roles	Active	Edit
Frank Liu	silanestest3@gmail.com	Yes	0412345678	School Admin	Yes	<a href="#">Edit</a>

## User Details

First Name:	<input type="text"/>	Last Name:	<input type="text"/>											
Email:	<input type="text"/>	Phone:	<input type="text"/>											
Roles:	School Administrator <input type="checkbox"/>													
Active:	<input checked="" type="checkbox"/>													
Teacher in:	Kinder	Prep	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	Year 7	Year 8	Year 9	Year 10	Year 11	Year 12
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="button" value="Save"/>		<input type="button" value="New User"/>												



**Data Input Status**

Year Level	Data Input Finalized
Prep	
Year 1	
Year 2	
Year 3	
Year 4	
Year 5	
Year 6	

**Subscription Status**

Service Type	Valid Until
<a href="#">Pay for one more year's subscription</a>	

**Payment History**

Invoice #	Total (AU\$)	Creation Date	Created By	Status	Included
-----------	--------------	---------------	------------	--------	----------

Figure 5 - School Admin page



NOTE: this fictional school will be used as example all through this user's manual. Later I will upload some generated student data into this school and do some allocations.

## Types of user roles

School users in Teacher's Pet can have two types of roles:

1. A school administrator role is able to add, deactivate or reactivate school administrators and teachers, and initiate payments for subscriptions.
2. A teacher role is able to perform class allocation in one or a number of year levels in the school.

A user can have any or both roles, and can be teachers in one or any combination of year levels.



As you can see in Figure 5 - School Admin page, after registering the fictional school, the author Frank Liu became the primary administrator of this school account.

The primary administrator has two special privileges that other administrators don't have:

1. No one can deactivate the primary administrator;
2. No one can make someone else the primary administrator except for the primary administrator him/herself.

## Add administrators and teachers

To add another school administrator or teacher, click the "New user" button in the "User Details" section:

The screenshot shows a 'User Details' form with the following fields and values:

- First Name: John
- Last Name: Smith
- Email: silanest4@gmail.com
- Phone: 0412345679
- Roles: School Administrator ☐
- Active: ☒
- Teacher in: A row of checkboxes for levels from Kinder to Year 12. The checkboxes for Year 2, Year 3, and Year 4 are checked.

At the bottom of the form are two buttons: 'Save' and 'New User'.

Figure 6 - Adding a new user

All fields will be cleared. Enter the new user's name, email and phone. If this user needs to be the school administrator role, check the "School Administrator" checkbox. If he/she needs to have teacher role in certain year levels, check the corresponding checkboxes. Then click the "Save" button.

In the above figure, the new user John Smith has been assigned teacher role in year 2, 3 and 4.

After saving, the "Users" section now becomes:



Users						
Name	Email	Email Verified	Phone	Roles	Active	Edit
Frank Liu	silanstest3@gmail.com	Yes	0412345678	School Admin	Yes	<a href="#">Edit</a>
John Smith	silanstest4@gmail.com	No	0412345679	Teacher in Year 2, Year 3, Year 4	Yes	

Figure 7 - Users section after adding second user

The system will send an activation email to the newly added user, the same as the one the author received when he registered the fictional school. After John activates his email, the "Email Verified" column of John's row in the above figure will become "Yes".

## Editing an existing user

An existing user cannot be deleted. He/she can only be edited or deactivated.

To edit an existing user, in the "Users" section, double click the row in the table with that user's name, or click the "Edit" link at the end of the row. For example, as the primary administrator of Frank's Fictional School, the author would like to perform the teacher role in all of the year levels of that school. So he double clicks his row in the "Users" section:

Users						
Name	Email	Email Verified	Phone	Roles	Active	Edit
Frank Liu	silanstest3@gmail.com	Yes	0412345678	School Admin	Yes	<a href="#">Edit</a>
John Smith	silanstest4@gmail.com	Yes	0412345679	Teacher in Year 2, Year 3, Year 4	Yes	<a href="#">Edit</a>

Figure 8 - Selecting an existing user to edit

That user's details will be shown in the "User Details" section:



**User Details**

First Name:	<input type="text" value="Frank"/>	Last Name:	<input type="text" value="Liu"/>																																					
Email:	<input type="text" value="silanestest3@gmail.com"/>	Phone:	<input type="text" value="0412345678"/>																																					
Roles:	School Administrator <input checked="" type="checkbox"/>																																							
Active:	<input checked="" type="checkbox"/>																																							
Teacher in:	<table><tr><td>Kinder</td><td>Prep</td><td>Year 1</td><td>Year 2</td><td>Year 3</td><td>Year 4</td><td>Year 5</td><td>Year 6</td><td>Year 7</td><td>Year 8</td><td>Year 9</td><td>Year 10</td><td>Year 11</td><td>Year 12</td></tr><tr><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr></table>												Kinder	Prep	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	Year 7	Year 8	Year 9	Year 10	Year 11	Year 12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kinder	Prep	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	Year 7	Year 8	Year 9	Year 10	Year 11	Year 12																											
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																											

Figure 9 - Editing an existing user

The author can then check all the year levels of his fictional school and click the “Save” button.

An administrator cannot deactivate him/herself, but he/she can deactivate another user, such as John Smith. To do this, just uncheck the “Active” checkbox in the above figure, and click “Save”.

As shown above, when editing an existing user, the user’s name or email cannot be changed. Following editing can be done:

- Changing the user’s phone number.
- Deactivating the user to prevent him/her from logging into the system, or reactivate him/her, by checking/unchecking the “Active” checkbox. Note that the primary administrator cannot be deactivated.
- Assign school administrator role to the user, or remove this role from the user, by checking/unchecking the “School Administrator” checkbox.
- Assign teacher role in any year level to the user, or remove the teacher role from any year level from him/her, by checking/unchecking the checkbox under the corresponding year level.

## Changing the primary administrator

To change the primary administrator of your school, the “change-to” user must have activated his/her account, be active, and be an administrator in the school account. Only the current i.e. old primary administrator can do this change.

To change the primary administrator, just change the email address in the school details section (see figure below) to the new primary administrator’s email, and click the “Save” button. No need to change the name or phone – they will be changed automatically after the “Save” button is saved.



School name: Frank's Fictional School

Street Address: Foo Bar Street

City:

Country: Australia

More address details:

School Type: ☒ Primary ☐ Secondary

**Primary Administrator**

First Name: Frank

Last Name: Liu

Email: sllanstest3@gmail.com

Phone: 0412-345-678

Active: ☒

Suburb: Melbourne

Postcode: 3000

State/Province: Victoria

Time zone: (UTC+10:00) Canberra, Melbourne, Sydney

Save

Figure 10 - Changing the primary administrator

## Paying for subscription

As discussed in the introduction chapter, the process of creating class lists has three steps:

Step 1: Pick your class-allocation criteria.

Step 2: Enter student data that are needed for the class-allocation process.

Step 3: Perform the class allocation

Without paying the subscription, step 1 and 2 can be done, but step 3 – allocating the classes cannot be done.

To pay for subscription, login as the school administrator, scroll down to the “Subscription Status” section on the “School Admin” page (see Figure 5 - School Admin page), and click button “Pay for one more year’s subscription”. Following dialog will be shown:

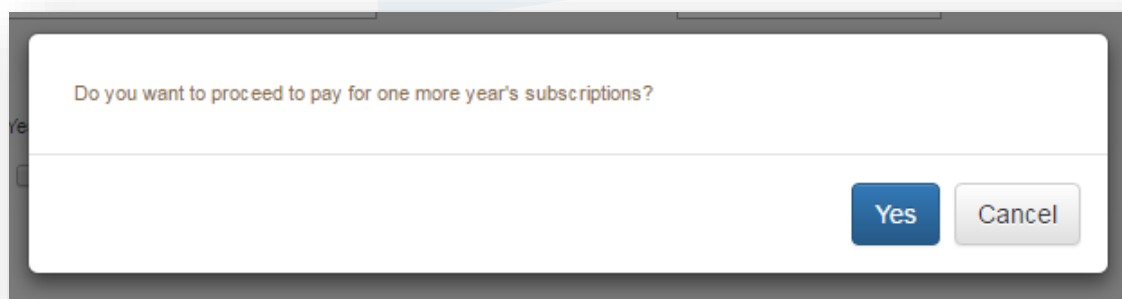


Figure 11 - Payment confirmation dialog

Click "Yes". Following "payment summary" page will be shown:

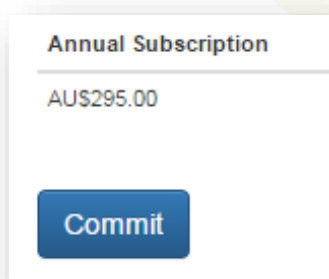


Figure 12 - Payment confirmation page

Once button "Commit" is clicked, the user will be asked to confirm again:

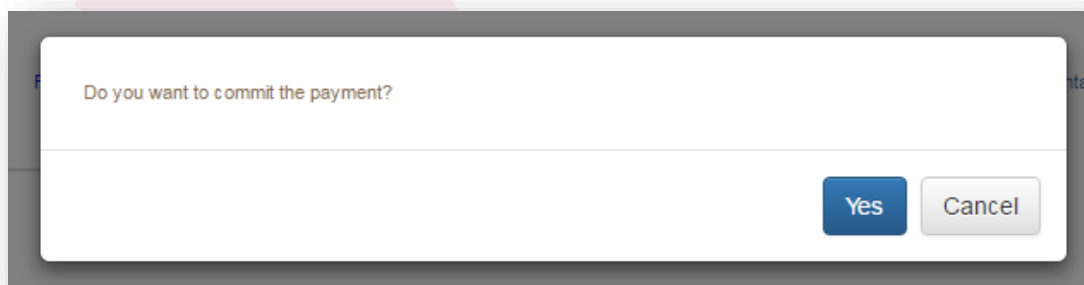


Figure 13 - Second payment confirmation dialog



Click "Yes", it will lead to a payment gateway, on which the user can pay the subscription using PayPal, Visa, Master or AMEX credit card. After payment goes through, refresh the "School Admin" page, the "Subscription Status" and "Payment History" section will become:

**Subscription Status**

Service Type	Valid Until
Class allocation	2018-01-17

Pay for one more year's subscription

**Payment History**

Invoice #	Total (AU\$)	Creation Date	Created By	Status	Included
449	AUS295.00	2017-01-17 06:14	Frank Liu	Paid	Class allocation:

Figure 14 - "School Admin" page after payment has gone through

## Selecting a role when logging in

To login, click the "Login" menu item. Following page will be shown:

Email

silanstest3@gmail.com

Password

\*\*\*\*\*

Log in

Forgotten password

Register new school

Figure 15 - Login page

After entering the correct email and password, if the user has a single role in the school account, he/she will be directed to the page that is most convenient for his/her role:





- If the role is school administrator, user will be directed to the "School Admin" page (see Figure 5 - School Admin page).
- If the role is a teacher, user will be directed to the "Pick Class-Alloc Criteria" page that will be discussed in the next chapter (see chapter "Step 1: Picking the Class Allocation Criteria").

If the user have multiple roles in the school account – for example the user is a school administrator and a teacher, or a teacher in multiple year levels, the following "Select Your Role" page will be shown:

You have multiple roles in your school. Please select one to proceed:

☐ Teacher of Year 2

☐ Teacher of Year 3

☐ Teacher of Year 4

Select role

Figure 16 - "Select Your Role" page

After selecting a role, the user will be directed to the page according to the selected role, as previously discussed.



## Step 1: Picking the Class Allocation Criteria

Right after the user logs in as a teacher, the step 1 page is displayed, showing all of the class-allocation criteria that is currently supported by the system:

Step 1: Pick the class allocation criteria

>> Step 2: Data Input

>> Step 3: Allocate Classes

Class-Allocation Task:

Creating classes for the new school year

Task Name:

Creating classes for the new school year

Select class allocation rules that you want to apply in allocating students to classes/groups:

Classing Rule	Description	Select Rule
Class by academic levels	All top-level students in one class, all medium-level students in one class, and all bottom-level students in one class.	<input type="checkbox"/>
Academic levels mixed	Each class has roughly the same number of top-level, medium-level and low-level students.	<input type="checkbox"/>
Behaviour	Spread students with different behaviours evenly across classes	<input type="checkbox"/>
Even class sizes	All classes have the same or very similar sizes.	<input type="checkbox"/>
Friends	Try to give each student as many friends as possible	<input type="checkbox"/>
Faith	Spread students with/without or with different faith evenly across classes	<input type="checkbox"/>
Separated by gender	Boys and girls must be separate, e.g. when allocating dormitories.	<input type="checkbox"/>
Gender balance	Each class has roughly same number of boys and girls.	<input type="checkbox"/>
Must include	Certain students must be in the same class - this takes much higher priority than the "Friends" rule	<input type="checkbox"/>
New students in class	Each class should have some new comers - we don't want to have the same classes as last year.	<input type="checkbox"/>
Secondary language	Students studying the same secondary language subject are better in the same class.	<input type="checkbox"/>
Special Need	Spread students with/without or with different special needs evenly across classes	<input type="checkbox"/>
Must exclude	Certain students must not be in the same class.	<input type="checkbox"/>

Specify the classes/groups into which you want to allocate the students for this task:

New class/group name:

Add class/group

[No class/group specified]

Save task

Figure 17 - The "Pick class allocation criteria" page



## Class-allocation criteria

The class-allocation criteria that are currently supported by the system are listed below:

Classing Rule	Description	Data You Need to Enter
Class by academic levels	All top-level students in one class, all medium-level students in one class, and all bottom-level students in one class.	Academic level of each student
Academic levels mixed	Each class has roughly the same number of top-level, medium-level and low-level students.	Academic level of each student
Behavior	Spread students with different behaviors evenly across classes	Behavior of each student
Even class sizes	All classes have the same or very similar sizes.	None
Friends	Try to give each student as many friends as possible	Five friends each student wants
Faith	Spread students with/without or with different faith evenly across classes	Faith of each student
Separated by gender	Boys and girls must be separate, e.g. when allocating dormitories.	Gender of each student
Gender balance	Each class has roughly same number of boys and girls.	Gender of each student
Must include	Certain students must be in the same class - this takes much higher priority than the "Friends" rule	Groups of students. Students in each group must be in the same class
New students in class	Each class should have some new comers - we don't want to have the same classes as last year.	Original class of each student
Secondary language	Students studying the same secondary language subject are better in the same class.	The secondary language subject each student selected
Special Need	Spread students with/without or with different special needs evenly across classes	Special need of the student if any
Must exclude	Certain students must not be in the same class.	Groups of students. Students in each group must NOT be in the same class

Table 1 - Class-allocation criteria

If your criteria is not listed here, we can add it into the system in only a few hours. See later section "What if my criterion is not listed?".

Different schools or even different grades in the same school may choose different sets of class-allocation criteria. Only pick those criteria that are relevant. The more criteria are selected on this step, the more data are need to be provided about each student in the next step, the more time is need to gather and enter the data.



Some of the criteria are mutually exclusive, in that they contradict to each other, and therefore can't be selected together. For example, the "Class by academic levels" and "Academic levels mixed" contradicts each other, so are "Separated by gender" and "Gender balance". When neither of the mutually exclusive criteria is selected, either one can be selected. Once one is selected, the other will become disabled and cannot be selected. So if the user changes his/her mind and wants to select the other one, he/she needs to first unselect the selected one, then the other one becomes available for selection.

After the criteria are selected, enter the names of the classes (or groups or dormitories) into which the students in the grade are to be allocated. Then click button "Save Task".

As shown in the figure below, the author logged in as year-three teacher John Smith in Frank's Fictional School, selected seven criteria, and created four classes for next year: 4A, 4B, 4C and 4D:



**Step 1: Pick the class allocation criteria** >> [Step 2: Data Input](#) >> [Step 3: Allocate Classes](#)

**Class-Allocation Task:** Creating classes for the new school year

**Task Name:** Creating classes for the new school year

Select class allocation rules that you want to apply in allocating students to classes/groups:

Classing Rule	Description	Select Rule
Class by academic levels	All top-level students in one class, all medium-level students in one class, and all bottom-level students in one class.	<input type="checkbox"/>
Academic levels mixed	Each class has roughly the same number of top-level, medium-level and low-level students.	<input checked="" type="checkbox"/>
Behaviour	Spread students with different behaviours evenly across classes	<input type="checkbox"/>
Even class sizes	All classes have the same or very similar sizes.	<input checked="" type="checkbox"/>
Friends	Try to give each student as many friends as possible	<input checked="" type="checkbox"/>
Faith	Spread students with/without or with different faith evenly across classes	<input type="checkbox"/>
Separated by gender	Boys and girls must be separate, e.g. when allocating dormitories.	<input type="checkbox"/>
Gender balance	Each class has roughly same number of boys and girls.	<input checked="" type="checkbox"/>
Must include	Certain students must be in the same class - this takes much higher priority than the "Friends" rule	<input type="checkbox"/>
New students in class	Each class should have some new comers - we don't want to have the same classes as last year.	<input checked="" type="checkbox"/>
Secondary language	Students studying the same secondary language subject are better in the same class.	<input checked="" type="checkbox"/>
Special Need	Spread students with/without or with different special needs evenly across classes	<input type="checkbox"/>
Must exclude	Certain students must not be in the same class.	<input checked="" type="checkbox"/>

Specify the classes/groups into which you want to allocate the students for this task:

New class/group name:

Add class/group

4A

4B

4C

4D

Save task

Figure 18 - Class-allocation criteria selected for year 3 in the fictional school



Four classes were created for year 3 of the fictional school, because the generated sample data contains 100 students and 25 students per class is typical. The sample data will be uploaded into the system in next step.

## What if my criterion is not listed?

Teacher's Pet supports any complex class-allocation criteria. Just for an extreme example, a teacher wants to have all students studying French in the same class, and all students studying Japanese evenly spread among all classes, and she doesn't care how students studying Indonesian spread or converge in the classes.

Obviously this class-allocation criteria is not currently in the system for selection, but all that teacher needs is to send the above requirement in a support request to Teacher's Pet. In two hours' time this criterion can be added into the system, then that teacher can start to do class allocation using this new criterion.

## What is a class-allocation task

The primary purpose of schools using Teacher's Pet is to allocate students into classes for next school year. There is a built-in task called "Creating classes for the new school year" for this purpose. It cannot be changed. When a school is registered, this task is already there for every year level of this school (see Figure 18 - Class-allocation criteria selected for year 3 in the fictional school). This special task is hereinafter called "the primary task".

However, schools can also use Teacher's Pet to create groups for other events, such as creating day groups and dormitories for a camping trip. The classes/groups into which students are allocated in each task are confined within the task – that is to say, the classes in one task are isolated from the classes in another task, and the allocation result in different tasks do not interfere with each other.

For example, teacher Melissa first has used Teacher's Pet to perform the primary task – to allocated the 100 students in year 3 into class 4A, 4B, 4C and 4D for next year. Then she creates a second task called "Camping – activity groups", and created ten activity groups called "G1" ~ "G10", and allocated the 100 students into them. During the day, students will engage in many activities in these groups, in which Melissa wants boys and girls mixed and balanced. Then Melissa creates a third task called "Camping – dormitories" with 18 dormitories called "D1" ~ "D18", and allocates students into them. These dormitories each have six bunker beds. Obviously girls and boys must not be in the same dormitory.

The class-allocation criteria picked by Melissa for the above three tasks are different. For the primary task – allocating students into classes for next year, she picks following criteria:

- Academic levels mixed
- Classes sizes
- Friends



- Gender balance
- Must include
- New students in class
- Must exclude

For the second task – allocating students into day activity groups in the camping trip, she picks following criteria:

- Class sizes
- Friends
- Must include
- Must exclude

For the third task – allocating students into dormitories in the camping trip, she picks following criteria:

- Class sizes
- Friends
- Separate by gender

All of the tasks that are created by the user – all tasks except for the primary task – are called hereinafter ‘secondary tasks’ or ‘custom tasks’.

## Cross-task data vs. task-specific data

As to the data needed by the system to do the class allocations, most of the data are the same across different tasks. For example, the gender of a student doesn’t change between tasks, and the friends each student wants don’t change either.

However, some data such as exclusions and inclusions can be different between different tasks.

For example, Peter and John are best friends and they always talk to each other during the lessons, so Melissa doesn’t want them to be in the same class. To achieve this, she adds them into a “must-exclude” group in the primary task (allocate classes for next semester). This will overwrite the friendship requirement between Peter and John, and make sure they are not in the same class. But for the camping trip, Melissa may want them together to have some fun, so she does not add them into a “must-exclude” group in the camping task. This way the friendship requirement will most likely get Peter and John into the same activity group.

In this example, the exclusion data about Peter and John is only visible in the primary task, but doesn’t exist in the camping task.

The system knows what data are cross-task and what data are task-specific:



Classing Rule	Data You Need to Enter	Task-Specific
Class by academic levels	Academic level of each student	No
Academic levels mixed	Academic level of each student	No
Behavior	Behavior of each student	No
Class sizes	None	--
Friends	Five friends each student wants	No
Faith	Faith of each student	No
Separated by gender	Gender of each student	No
Gender balance	Gender of each student	No
Must include	Groups of students. Students in each group must be in the same class	<b>Yes</b>
New students in class	Original class of each student	<b>Yes</b>
Secondary language	The secondary language subject each student selected	No
Special Need	Special need of the student if any	No
Must exclude	Groups of students. Students in each group must NOT be in the same class	<b>Yes</b>

Table 2 - Data needed for class-allocation criteria - are they task specific or not

For those cross-task data, when it is entered in one task, it appears in all other tasks. Teachers usually enter all students and their cross-task data in the primary task, and simply use these data in other tasks. Those task-specific data need to be entered in every task that needs it – the value that is entered for a certain criterion of a certain student in task X is invisible in task Y, and does not interfere with the value entered in task Y.

## Creating a new task

As said previously, the primary task is built-in. It is there for every year level of every school, so users don't need to create it. Any other secondary/custom task needs to be created by the user. To do so, on the "pick class-allocation criteria" page (see Figure 17 - The "Pick class allocation criteria" page), select "[New]" on the "Class-Allocation Task:" dropdown box. The "Task Name" textbox will become editable. Enter a name of the user's own choice. Then select the class-allocation criteria and add class/group names. This process is identical to the process discussed in section "Class-allocation criteria".





## Step 2: Entering Student Data

### All students are displayed in a table

The criteria that the user has selected in step 1 (Figure 17 - The “Pick class allocation criteria” page) determines what data must be entered for each student. Once step 1 is finished, click the “Step 2: Data Input” link, the “step 2 – data input” page is displayed, on which the user enters data about the students.

Following is what is displayed when the author logged in as year-three teacher John Smith in Frank’s Fictional School for the first time:

Step 1: Pick the class allocation criteria >> **Step 2: Data Input** >> Step 3. Allocate Classes

**Class-Allocation Task:** Creating classes for the new school year ▼

[Upload Data](#) [Unfinalize](#) ⓘ [New Student](#) [Exclusions/Inclusions](#) [Pick My Friends](#) [Download Data](#)

Students (0)	Email	Current Class	Academic Level	Gender	Secondary Language	Desired Friends	Include	Delete	Edit
--------------	-------	---------------	----------------	--------	--------------------	-----------------	---------	--------	------

Figure 19 – The “Enter student data” page without any student in my fictional school example

All students are displayed on a table, with each student’s data being a row in that table. In the above figure, this table is empty, since no student has been entered into the system. It will look like the following when there are some students in the system:























Upload Data		Finalize		New Student							Exclusions/Inclusions		Pick My Friends		Download Data	
Students (100)	Email	Current Class	Academic Level	Gender	Secondary Language	Desired Friends	Include 		Delete 		Edit					
ALISE COLMENERO	silanliu@gmail.com	1	3	OTHER	CHINESE	JOAQUIN ELKIN, TYREE BRECHT, MELISSIA MCGLOTHLEN, SHANIKA LAMOREAUX, ISA ETHERTON					Edit					
ARDELLA ALBERTSON	silanstest@gmail.com	1	2	F	CHINESE	HILARIO MCLARTY, BERNIE REGINA, NATHANAEL ENTREKIN, ARTIE FOBBS, DONELLA LAZARD					Edit					
BRAIN BAZEMORE	silanstest2@gmail.com	1	3	M	FRENCH	SANTOS YIP, TREY COPAS, SUNNI STANGL, GINA MINEAR, VELMA MCMILLIAN					Edit					
CECILY REESE	[test]	1	2	F	CHINESE	TREY COPAS, ARIEL PUCCI, TOWANDA BANGS, GEORGIANA VANCELEAVE, SHANTA OBERMILLER					Edit					

Figure 20 – The students table in the fictional school example

In this table, the first three columns are always the same, no matter what class-allocation criteria are selected in step 1:

1. The first column is the student's name. Data for this column must be provided.
2. The second column is email of the student. Email is not mandatory if it is not needed – it is only needed if the teacher intends to invite the students to pick their own friends on Teacher's Pet portal. This will be discussed in later section "Students picking their own friends". If email is not entered, this column will be blank.
3. The third column is the student's current class. Data for this column must be provided.

The last three columns of the student table are not used to display data. They are for actions that the user is allowed to perform against each student record:

1. Include: a checkbox representing whether the student is included in the class-allocation. See section "Excluding a student from a custom/secondary task" for details. If you check or uncheck the checkbox in the column header (not in the student row), then the checkboxes in all student rows will be checked or unchecked. Once you have finished checking or unchecking, just click the floppy disk icon in the column header to save your changes.
2. Delete: a checkbox used to delete a student and all his/her data. You can select multiple students and delete them all at once. If you check or uncheck the checkbox in the column header, then all the checkboxes in all student rows will be checked or unchecked. Once you have selected all students to be deleted, just click the "X" button in the column header to delete all the selected students.
3. Edit: a link to click to edit the student's data.

In between the first three columns and last three columns, are those columns containing the student data that are needed by the system to do class allocation. They are needed because certain class-allocation criteria were selected in step 1 (Figure 17 - The "Pick class allocation criteria" page). In the fictional school example, as shown in Figure 20 – The students table in the fictional school example, they are



- Academic Level: since criterion “academic levels mixed” was selected (see Figure 18 - Class-allocation criteria selected for year 3 in the fictional school).
- Gender: since criterion “gender balance” was selected.
- Secondary Language: since criterion “secondary language” was selected.
- Friends: since criterion “friends” was selected.

Altogether seven criteria were selected, the other three are

- Even class size: this criterion doesn’t need any data.
- New students in class: this criterion also doesn’t need any data.
- Must exclude: this data is not displayed in the table because it is not about a single student, instead it is a group of students that shouldn’t be in the same class. This grouping is displayed when the “Exclusions/Inclusions” link is clicked, which is on the top-right corner in Figure 20 – The students table in the fictional school example. It will be discussed in later section “Inclusions & exclusions”.

If the author goes back to step 1 (Figure 17 - The “Pick class allocation criteria” page) and unselects criterion “academic levels mixed”, then come back to this step 2 page, the “Academic Level” column will disappear from the students table, because the system no longer need this data to do the class allocation.

## Entering new students

### *Two ways to enter students*

There are two ways of entering students into the system:

1. Manually entering students one by one;
2. Have all students and their data in student data CSV files, and uploading them in one go.

### *Manually entering a new student*

To enter a new student manually, click the “New Student” link on the “Enter student data” page:



Step 1: Pick the class allocation criteria >> **Step 2: Data Input** >> Step 3: Allocate Classes

**Class-Allocation Task:** Creating classes for the new school year ▼

[Upload Data](#) [Unfinalize](#) [New Student](#) [Exclusions/Inclusions](#) [Pick My Friends](#) [Download Data](#)

Students (0)	Email	Current Class	Academic Level	Gender	Secondary Language	Desired Friends	Include	Delete	Edit
							<input type="checkbox"/>	<input type="checkbox"/>	

Figure 21 – The “New Student” link on the “step 2 – data input” page

A hidden section on the same page will appear:



[Upload Data](#) [Finalize](#) [New Student](#) [Exclusions/Inclusions](#) [Pick My Friends](#)

### Student Details

[Hide](#)

**Note: the combination of the student's first name, middle name and last name must be unique in your school.**

First name:

Middle name:

Last name:

Email:

Current Class:

[Select] ▼

[Add or delete items in this drop-down](#)

Academic Level:

[Select] ▼

[Add or delete items in this drop-down](#)

Gender:

[Select] ▼

[Add or delete items in this drop-down](#)

Secondary Language:

[Select] ▼

[Add or delete items in this drop-down](#)

### Desired Friends

[No student selected] ▲

▼

Save student

Generate PIN

Send "Pick My Friends" email

Get PIN

Figure 22 - Creating a new student

The above figure is what is shown in the fictional school example. The input fields in the above figure (drop down boxes and text boxes) has a one-to-one correspondence with the columns shown in the student table (Figure 20 – The students table in the fictional school example). If the author goes back to step 1 (Figure 17 - The “Pick class allocation criteria” page) and unselect criterion “academic levels mixed”, then come back to the page shown in above figure, the “Academic Level” text box will disappear, because the system only asks the user to enter data that it needs to do the class allocation.



All input fields are mandatory except for the “friends” list box. The reason these input fields are here is because the corresponding class-allocation criteria had been selected in step 1 (Figure 17 - The “Pick class allocation criteria” page). If no data is available for a certain input field, then perhaps the corresponding class-allocation criterion shouldn’t have been selected. Therefore, the text boxes must not be left empty, and an item must be selected on each dropdown box, otherwise the system will refuse to save the student.

The only exception is the “friends” list box: a student can be saved without selecting any friend. This is because, for example, when entering the first student in the year level, none of the friends this student wants would have been entered into the system.

After student data has been supplied on the form, click the “Save student” button. That student will appear in the student table (see Figure 20 – The students table in the fictional school example).

### *Defining options for each dropdown box*

In the fictional school example shown in Figure 22 - Creating a new student, because it is the first time the author login as the year-three teacher John Smith, the dropdown boxes in the above figure wouldn’t have contained any option/item. Before he could enter data for any student, he needs to add options into the dropdown boxes so that he can select one of them to finish the date entry.

It is important to know that the system does not have a set of built-in options for each dropdown box. Instead the system requires the users to define their own set of values for each dropdown box. This is not a hassle but a nice feature, because, for the same dropdown box, different schools or even different grades in the same school may want to use different set of values. For example, in one school, academic levels may be represented by A, B, C, D and E, while in another school it may be represented by 1, 2 and 3. Another example is the “faith” criteria. One school may want to have “yes” and “no” as the options for “faith”, while another school may want to have “Christian”, “Catholic”, “Buddhism” etc. as the options.

So each school grade defines their own set of options for each dropdown box. Such options are invisible to other schools and other grades in the same school.

If students are to be manually entered into the system, then the options for the dropdown boxes must be defined before the first student is entered. The options of each dropdown box is defined only once. Once they are defined, any teacher in the same grade will be able to see and choose those items in the dropdown box.

If the student data are contained in CSV data files to be uploaded into the system, then there is no need to add options into the dropdown boxes, because those options will be picked up from the data files during the importation process.



For example, if the data files contain "1", "2" and "3" in the "Academic Level" column, then, after the importation, the "Academic Level" dropdown box will contain option "1", "2" and "3". If the data files contain "A", "B", "C", "D" and "E" in the "Academic Level" column, then after importation the dropdown box will contain option "A", "B", "C", "D" and "E".

Suppose before the data importation, the "Academic Level" dropdown box already contains option "1", "2" and "3". Then, a data file that has "A", "B", "C", "D" and "E" in the "Academic Level" column is uploaded. After the importation, the "Academic Level" dropdown box will contain option "1", "2", "3", "A", "B", "C", "D" and "E".

Items can be added into the dropdown boxes or deleted from them at any stage. If an item such as "E" is deleted from the "Academic Level" dropdown box, students that have their "Academic Level" property being "E" will not be affected. In other words, the system doesn't retrospectively change the student data when items in the dropdown box is changed.

In the fictional school example, to define the options in the "current class" dropdown box, click the "Add or delete items in this dropdown" link which is located to the right of the "current class" dropdown box (see Figure 22 - Creating a new student). Since no item has been defined, following dialog will be shown:

The dialog box has a title bar with the text "Add or delete items in the "Current Class" drop-down". Below the title bar is a text input field with the placeholder text "Add new option:". To the right of the input field are three buttons: "Add new option" (blue), "Delete selected option" (blue), and "Close" (grey).

Figure 23 - Add or delete items in the dropdown box - when no item is defined

In the fictional school example, the students are currently in four classes called "3A", "3B", "3C" and "3D". So the author enters "3A" in the textbox, then click button "Add new option". Then he enters "3B" and click the same button, and so on. After entering the four current classes, the dialog becomes following:



Add or delete items in the "Current Class" drop-down

Select one of the following items to delete:

☐ 3A

☐ 3B

☐ 3C

☐ 3D

Add new option:

Add new option Delete selected option Close

Figure 24 - Add or delete items in the dropdown box - when there are items defined

To delete an item, click the corresponding radio button, then click the "Delete selected option" button. That option will disappear.

After all four current classes have been entered, the author clicks the "Close" button. This dialog will disappear. Now the options that have been added appear in the "current class" dropdown:





Figure 25 - Items added into the dropdown box

## Editing existing students

To edit an existing student, double-click the student row in the student table (see Figure 20 – The students table in the fictional school example), or click the “Edit” link in the last column of that student row. The same dialog shown in Figure 22 - Creating a new student will be shown, except that the student data are prefilled on the form and the student name textboxes are read-only:



Upload Data   Unfinalize   New Student   Exclusions/Inclusions   Pick My Friends   Download Data

### Student Details

Hide

**Note:** the combination of the student's first name, middle name and last name must be unique in your school.

First name:

Middle name:

Last name:

Email:

Current Class:  [Add or delete items in this drop-down](#)

Academic Level:  [Add or delete items in this drop-down](#)

Gender:  [Add or delete items in this drop-down](#)

Second Language:  [Add or delete items in this drop-down](#)

### Desired Friends

Each student gets to specify a number of friends to be in his/her class

Add a friend by name, then hit enter:

Friends already added:

HILARIO MCLARTY

BERNIE REGINA

NATHANAEL ENTREKIN

ARTIE FOBBS

DONELLA LAZARD

Figure 26 - Editing an existing student

Apart from the student's name, everything else about this student can be changed. After changes are made, click the "Save student" button.

## Deleting students

There are two scenarios in which the teacher deletes students:

1. A student leaves the school.



2. The students who were entered into the system last year as the current grade are now in the higher grade, and the students that are currently in this grade need to be entered into the system.

To delete one, some or all of the students in the current grade, see section “All students are displayed in a table”.

## Friends

### *Three ways to enter friends*

One of the biggest value that Teacher’s Pet brings to the schools, is that it can keep friends together in the new classes, while catering for lots of other class-allocation criteria. There are three ways of entering friends into the system:

1. Ask each student to submit to the teacher a list of five friends, then the teacher manually enters the friends into the system on the Teacher’s pet portal.
2. Have the students pick their own friends on the Teacher’s pet portal.
3. Have the friends in the student data CSV file, and upload that CSV file in a few mouse clicks.

### *Manually entering friends (by teacher)*

When entering a new student (Figure 22 - Creating a new student), or editing an existing student (Figure 26 - Editing an existing student), friends of this student can be added or removed on the “Desired Friends” section.

To add a new friend, just enter a search keyword using the first few characters of the student’s first name, middle name or last name, case-insensitive, into the “Add friend by name” textbox, then hit “Enter” on the keyboard. On certain browsers such as Internet Explorer, hitting the “Enter” key doesn’t work, so the user needs to click outside of the textbox, for example on a blank spot on the page.

For example, if “Daniel Frisk” is to be entered as a friend, the user can simply enter “dan” or “dani” or “fri”. If what is entered matches only a single student in the whole year level, that student is immediately added into the “Friends already added” list box. If the search keyword matches multiple students – for example if “dan” is entered and the year level has “Daniel Frisk”, “Danille Franzoni” and “Ena Dangelo”, then an extra dropdown box will appear between the textbox and the list box, and all the matching students will be added into that dropdown box, allowing the user to select the one:



Each student gets to specify a number of friends to be in his/her class

Add a friend by name, then hit enter:

Multiple results found. Please select:

Friends already added:

SANTOS YIP  
TREY COPAS  
SUNNI STANGL

[Please select]  
[Please select]  
DANIEL FRISK  
DANILLE FRANZONI  
ENA DANGELO

Figure 27 - Search for friend by name - when there are multiple matches

The more characters are entered as the search keyword, the narrower the search result, the bigger chance a single student is returned, avoiding the need to select from a drop down box.

To delete an existing friend, click the friend in the list box, the “Delete friend” link will appear. Click it to delete the selected friend:

**Desired Friends**

Each student gets to specify a number of friends to be in his/her class

Add a friend by name, then hit enter:

Friends already added:

SANTOS YIP  
TREY COPAS  
SUNNI STANGL  
GINA MINEAR  
VELMA MCMILLIAN

Delete selected

Figure 28 - How to delete a friend

The user can hold the “Ctrl” key on the keyboard and select multiple friends in the list box, and delete them all at once.



Compared with having to type the full names of friends in Excel, the capability to select a friend by entering only the first few characters of his/her name makes it much quicker to enter friends, and eliminates the possibility to bring typos into the system.

For young students who are not fluent with web browsers, this is usually the only method of entering friends for them.

### *Students picking their own friends*

As shown in the previous section, entering friends one-by-one manually on the Teacher's Pet portal is actually quite quick and reliable in comparison with total manual data entry. However, it is still a task of hours. If all the students have access to Internet and are fluent with web browsers, it is much quicker to invite them to pick their own friends on Teacher's Pet portal.

To invite all students in the year level to pick their own friends, click the "Pick My Friends" link at the top of the page:

Step 1: Pick the class allocation criteria >> **Step 2: Data Input** >> Step 3: Allocate Classes

**Class-Allocation Task:** Creating classes for the new school year

[Upload Data](#) [Unfinalize](#) [New Student](#) [Exclusions/Inclusions](#) **[Pick My Friends](#)** [Download Data](#)

Students (0)	Email	Current Class	Academic Level	Gender	Secondary Language	Desired Friends	Include	Delete	Edit
--------------	-------	---------------	----------------	--------	--------------------	-----------------	---------	--------	------

Figure 29 - The "Pick My Friends" link

The following section that is normally hidden will appear on the same page:



[Upload Data](#) [Unfinalize](#) [New Student](#) [Exclusions/Inclusions](#) [Pick My Friends](#) [Download Data](#)

## Invite Students to "Pick My Friends" Hide

There are two ways you can invite students to "Pick My Friends" directly on Teacher's Pet. Either way, first you need to click the "Generate PINs" button to generate pins for all students enrolled in this 'Creating classes for the new school year' task, which they can use to log into Teacher's Pet. The PINs will remain valid for seven days.

**Generate PINs for all students enrolled in this task**

After you have generated the PINs, you can choose one of the two following methods to get students to pick their own friends:

**1. Easiest and safest - if every student has an email:**

The system sends an invitation email to every student enrolled in this "Creating classes for the new school year" task. The student clicks the "Pick my friends" link in the email. It goes to the "Pick My Friends" page on Teacher's Pet. No login is required, yet it is secure.

**Send emails to all students enrolled in this task**

If anyone doesn't received this email, you can double-click the student row below and resend the email only to this student. Or you can manually add the friends for him/her.

**2. If students don't have emails:**

Display the PINs of all students. Print them out, and give each student his/her own PIN. Tell them don't shown anyone their own PINs. Ask every student to login and enter the school name, his/her own name, and the pin. Then they will be directed to the "Pick My Friends" page.

Please be prepared for some "teach-support", because some students may make typos and panic and ask for your help.

**Get student PINs**

Figure 30 - Inviting students to pick their own friends

As shown in the above figure, there are two ways to let students pick their own friends, but either way, a PIN must first be generated for each student. The PINs will be valid for seven days. Just click the "Generate PINs for all students enrolled in this task" button.

If, for example, the user is doing the class allocation for a camping trip and not all students in the year level are enrolled in this camping trip (see "Excluding a student from a custom/secondary task"), then only the enrolled students will have their PINs generated.



After the PINs have been generated, if the students all have their own emails in the system, the user should click the “Send emails to all students enrolled in this task” button. Each student will get the following email:

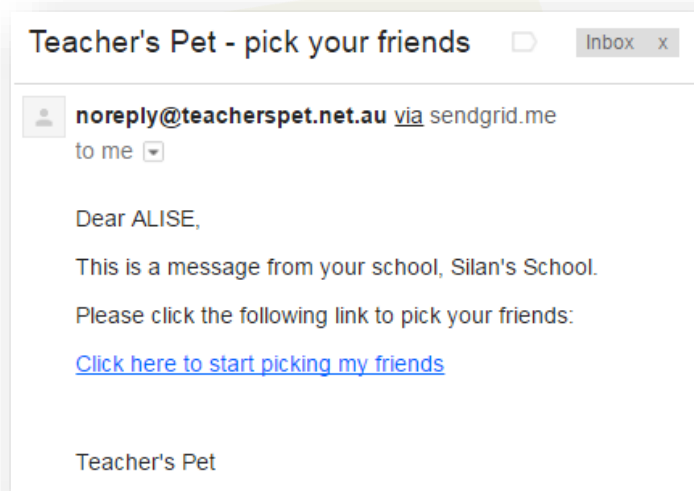


Figure 31 - The invitation email to students to pick their own friends

When the student clicks the “Click here to start picking my friends” link in the email, he/she will be brought to the Teacher’s Pet portal, and see the following page:



## Pick your friends

[Creating classes for the new school year]

Who do you want to be with in the new class?

Hi ALISE, please enter your friend's name, such as "John" or "John Smith", then hit "Enter" or click outside the textbox.

Friends that I have already picked:

JOAQUIN ELKIN  
TYREE BRECHT  
MELISSIA MCGLOTHLEN  
SHANIKA LAMOREAUX  
ISA ETHEERTON

Save Friends

Figure 32 - A student picking his/her own friend on the portal

There is no login process in which the student enters his/her name/ID and password. This is because the system encrypted the student's identity and PIN in the link in the email. No one can fake this email and impersonate this student.

The way the student enters the first few characters of a friend's name to search for the friend, and the way a dropdown box is shown when multiple matches are found, are identical to the teacher entering friends for the students in section "Manually entering friends". All students in the grade can finish picking their own friends within half an hour, with each student taking two to five minutes.

If the students don't have their own emails, but they do have access to Internet, a less ideal way to invite them to pick their own friends is to click the "Get student PINs" button in Figure 30 - Inviting students to pick their own friends. Following dialog will pop up:





Website	https://www.teacherspet.net.au/student/pick-my-friends
School ID	30
Student Name	<b>ALISE COLMENERO</b>
PIN	1-3495
Website	https://www.teacherspet.net.au/student/pick-my-friends
School ID	30
Student Name	<b>ARDELLA ALBERTSON</b>
PIN	1-2912
Website	https://www.teacherspet.net.au/student/pick-my-friends
School ID	30
Student Name	<b>BRAIN BAZEMORE</b>
PIN	1-1471
Website	https://www.teacherspet.net.au/student/pick-my-friends
School ID	30
Student Name	<b>CECILY REESE</b>
PIN	1-5410

Figure 33 - Student login details to pick their own friends

The teacher can print it out, cut the print out into pieces and give each piece to the corresponding student. The student will need to open a web browser and enter the website URL. Then the following student login page will be shown:

### Login

School ID	<input type="text"/>
Your first name	<input type="text"/>
Your middle name	<input type="text"/>
Your last name	<input type="text"/>
The PIN teacher gave you	<input type="text"/>
<input type="button" value="Log in"/>	

Figure 34 - Student login page to pick his/her own friends



The student must enter the school ID, his/her name and PIN that are shown on the piece of paper, to be able to login, and see the page shown previously in Figure 32 - A student picking his/her own friend on the portal. This solution is not as simple as the first one which sends invitation emails to the students. A student may have a typo when logging in, and panic, and the teachers may get a lot of requests for help, especially if the students are in the first few years in school.

## Inclusions & exclusions

Inclusions and exclusions data are different from the student data that has been discussed so far, because they are not about any particular student, but a group of students. Friends are also about a particular student because they are wanted by this student.

Exclusions: any two students in the same exclusion group must not be in the same class. An exclusion group usually contains two students but it can contain any number of students.

Inclusions: all students in the same inclusion group must be in the same class. Again an inclusion group can contain any number of students.

Exclusions and inclusions are given much higher priority by the system than friendship, so they will always override friendship requirements if there is a conflict.

For example, John and Peter are best friends, so they picked each other when they picked their friends. However, the teacher doesn't want them to be in the same class because they frequently talk to each other during the lessons. In this case the teacher does not need to remove John and Peter from each other's friend list. All he/she needs to do is to add them into an exclusion group. The system will then make sure they are not in the same class.

### *Two ways to enter inclusions & exclusions*

There are two ways of entering the exclusion and exclusion groups:

1. Manually enter the exclusion and exclusion groups on the portal;
2. Have all exclusion and exclusion groups in a CSV file, and upload this file in one go.

### *Manually entering inclusions & exclusions*

To manually enter inclusions and exclusions, click the "Exclusions/inclusions" link on the "Step 2 – Data Input" page:



Step 1: Pick the class allocation criteria >> **Step 2: Data Input** >> Step 3: Allocate Classes

**Class-Allocation Task:** Creating classes for the new school year ▼

[Upload Data](#) [Unfinalize](#) ⓘ [New Student](#) **[Exclusions/Inclusions](#)** [Pick My Friends](#) [Download Data](#)

Students (0)	Email	Current Class	Academic Level	Gender	Secondary Language	Desired Friends	Include	Delete	Edit
							<input type="checkbox"/>	<input type="checkbox"/>	

Figure 35 - The "exclusions/inclusions" link

The following section that is normally hidden will be shown on the same page:

[Upload Data](#) [Unfinalize](#) ⓘ [New Student](#) **[Exclusions/Inclusions](#)** [Pick My Friends](#) [Download Data](#)

## Exclusions and Inclusions Hide

### Create a new exclusion/inclusion group

[Select] ▼

Enter student name, then hit "Enter" or click elsewhere:

students in the group:

[No student selected]

▲ ▼

[Save](#)

Figure 36 - The "exclusions and inclusions" section on the "step 2 – data input" page



The type of the group (inclusion or exclusion) must be selected on the dropdown box at the top. Then a student can be searched and added into or removed from the group in exactly the same manner as friends are added or removed in section “Manually entering friends (by teacher)”.

After you have selected the type of group and added a few students into the group, the above figure becomes the following:

The screenshot shows a web application interface with a top navigation bar containing links: 'Upload Data', 'Unfinalize', 'New Student', 'Exclusions/Inclusions' (which is underlined), 'Pick My Friends', and 'Download Data'. The main content area is titled 'Exclusions and Inclusions' in blue text, with a 'Hide' link to its right. Below the title is the instruction 'Create a new exclusion/inclusion group'. There is a dropdown menu set to 'Must NOT be in the same class'. Below this is a text input field with the placeholder 'Enter student name, then hit "Enter" or click elsewhere:'. Underneath the input field is the label 'students in the group:' followed by a list box containing two names: 'ALISE COLMENERO' and 'DANIEL FRISK'. At the bottom left of the main content area is a blue 'Save' button.

Figure 37 - An exclusion group with two students

Then click the “Save” button to save the group.

After a few groups have been created, the user interface becomes the following:



Upload Data   Unfinalize ⓘ   New Student   Exclusions/Inclusions   Pick My Friends   Download Data

## Exclusions and Inclusions Hide

### Create a new exclusion/inclusion group

Must NOT be in the same class ▼

Enter student name, then hit "Enter" or click elsewhere:

students in the group:

[No student selected] ▲ ▼

Save

### Must NOT be in the same class

Any two of the students in this group shall not be in the same class.

Enter student name, then hit "Enter" or click elsewhere:

students in the group:

KIETH POEHLER  
NATHANAEL ENTREKIN ▲ ▼

Save group   Delete group

### Must NOT be in the same class

Any two of the students in this group shall not be in the same class.

Enter student name, then hit "Enter" or click elsewhere:

students in the group:

HASSAN BALKE  
WARD LEDERER ▲ ▼

Save group   Delete group

Figure 38 - Exclusion groups

To add a new group, always use the group of controls on the top of the above figure. You can also add students into an existing group, or delete members from an existing group, in the same way as you add or remove friends in section "Manually entering friends (by teacher)".



## Uploading student data file (CSV)

If your school uses a school management software system, most of the data needed by Teacher's Pet to do the class allocation, such as student name, gender, email and academic level, may have already existed in the school management system. In this case, the student data can be exported into a CSV (comma separated values) file. This file must be renamed to what is required by Teacher's pet (see discussion below), then opened using Microsoft Excel. Unneeded columns can be removed, needed columns must be renamed as required by Teacher's Pet. Then this CSV file can be uploaded into Teacher's Pet in a few clicks.

Allowing the users to upload data in a file is the quickest way to get data into the system, however it is always tricky, since there are many things that can go wrong, such as the file name and the column names. Teacher's Pet has a smart feature that takes all these risks away.

Once the class-allocation criteria have been selected, on the "step 2 – data input" page, click the "Upload Data" link:

Step 1: Pick the class allocation criteria >> **Step 2: Data Input** >> Step 3: Allocate Classes

**Class-Allocation Task:** Creating classes for the new school year

**Upload Data** Unfinalize New Student Exclusions/Inclusions Pick My Friends Download Data

Students (0)	Email	Current Class	Academic Level	Gender	Secondary Language	Desired Friends	Include	Delete	Edit
							<input type="checkbox"/> +	<input type="checkbox"/> X	

Figure 39 - The "Upload Data" link on the "Student Data" page

The following section that is usually hidden will appear on the same page:



Upload Data Finalize New Student Exclusions/Inclusions Pick My Friends

Upload student data file(s) Cancel Show me the data file format Download sample data files

Choose File No file chosen

Upload files

Figure 40 - The "upload student data" section on the "student data" page

Click the "Show me the data file format" link on the above figure, a dialog will pop up, showing exactly what data files are needed and what is the exact content in each of those data files using sample data. In the fictional school's example, when the link is clicked, the following dialog is shown:

File Name: "Year 3 - Creating classes for the new school year - student data.csv"

First Name,Middle Name,Last Name,Current Class,Academic Level,Gender,Secondary Language,Desired Friends  
ARDELLA,,ALBERTSON,A,1,M,French,"ELLIOT HELFRICH,CHONG FLOURNOY,ADRIANA MCKITTRICK,WAI TILLISON,DARLINE NICHOLAS"  
BRAIN,,BAZEMORE,B,2,F,Chinese,"JARRETT DEPAOLI,HASSAN BALKE,TAMEKIA TIMMINS,DONELLA LAZARD,OPHELIA LETOURNEAU"  
...

File name: "Year 3 - Creating classes for the new school year - student exclusions and inclusions.csv"

Exclusion or Inclusion,Students  
Must NOT be in the same class,"DANILLE FRANZONI, VELVA LITCHFIELD"  
Must NOT be in the same class,"BELIA CORTESE, GUILLERMO GAULIN"  
...

OK

Figure 41 - Student data CSV file format



This dialog shows that, based on the class-allocation criteria that have been selected on step 1 (Step 1: Picking the Class Allocation Criteria), two CSV files are needed for the system to do the class allocation:

- Year 3 - Creating classes for the new school year - student data.csv
- Year 3 - Creating classes for the new school year - student exclusions and inclusions.csv

The required file names has three parts separated by two dashes (" - "):

1. The year level, in our example it is "Year 3".
2. The task name, in our example it is "Creating classes for the new school year".
3. The type of the file – the first file is "student data", which contains all data about each student – name, gender, email, academic level, etc. The second file is "student exclusions and inclusions", which contains groups of students that must/must not be in the same class.

The data files that the user actually uploads must have exactly the same as recommended above, or start with exactly the same names followed by arbitrary suffix.

For example, if what is required is

Year 3 - Creating classes for the new school year - student data.csv

The actual file can be named

Year 3 - Creating classes for the new school year - student data – Frank's first try.csv

On the above dialog, under each file name, there is a box which contains sample file content. If the actual CSV data files are opened with a text editor, they must resemble the sample content.

Compare the columns in the first CSV file with the data columns in the student table shown in Figure 20 – The students table in the fictional school example, it becomes obvious that they are the same set of columns. Likewise, what is contained in the second CSV file are the exclusion and inclusion groups that can be entered manually on the portal as shown in Figure 36 - The "exclusions and inclusions" section on the "step 2 – data input" page.

Therefore, this dialog tells exactly what data files are needed, what should their names be, and what columns are needed in the data files, based on what class-allocation criteria that have been selected on step 1. So this is not a static dialog. What is shown on this dialog is dynamically generated from the criteria selected on step 1. This takes all the uncertainties away from the users when they are preparing the data files.

For example, if the user goes back to step 1 and untick the "Academic Levels Mixed" criterion and comes back and click the "Show me the data file format" link again, the "Academic Level" column would disappear from the sample data file. Likewise, if the user unchecks the "Must Exclude" criterion, the second CSV file would disappear altogether.





An alternative to showing the sample data files in the dialog (Figure 41 - Student data CSV file format), is to click the "Download sample data files" link shown in Figure 40 - The "upload student data" section on the "student data" page, to actually download the sample data files onto the user's PC.

Some important facts to know about uploading student data files:

- The names of the actual CSV files which the user wants to upload must either be the same as what is required, or start with what is required with an arbitrary suffix.
- The column names must be the same as the generated columns.
- The sequence of columns does not matter.
- If a data column in the student data file corresponds to a dropdown box in the "student details" form (Figure 22 - Creating a new student), then the data values contained in this column will be inserted into that dropdown box. See section "Defining options for each dropdown box" for details.
- The CSV files may contain other columns that are not required. They will simply be ignored so will not cause error. However, it is recommended that the data files to upload do not contain too many unneeded columns, since it increases the upload and processing time.
- The CSV files do not have to contain all required columns. Some required columns missing will not cause error. The missing data can later be manually entered on the web interface. Typically, the CSV file exported from the school management system wouldn't have contained the friend requests. It can still be uploaded. Then the teacher can send invitation emails to the students to invite them to pick their own friends on the portal, or manually enter friends for the students on the portal.
- The same data file can be uploaded repeatedly. The second time and so on will not change anything in the system, and there will not be any duplication of data created. However, if, after the initial upload, changes have been made, then the same data files are uploaded again, the changes that have been made will be lost. Therefore, it is recommended that every time manual data changes have been made on student data and grouping, download the changed data into CSV files (see next section on how to download) to replace the original data files (copy the original data files into an archive folder).

In the fictional school's example, once the two required CSV files are ready (see section "Sample student data" in the appendix for the content of the two files used in the fictional school example), the author clicks the "Choose File" button to browse for the files. After the two files have been selected, the "upload data" section becomes the following:

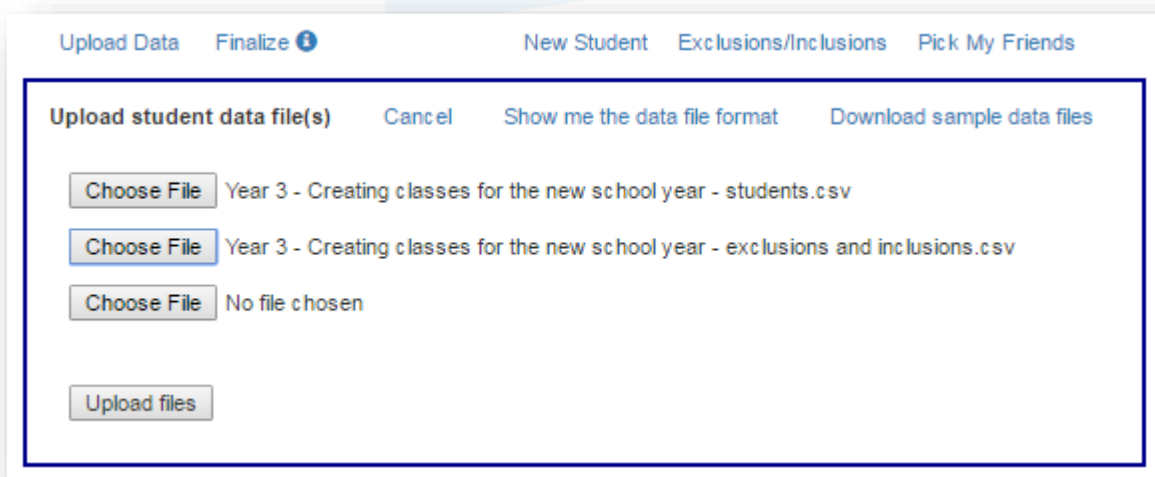


Figure 42 - The "upload data" section after data files selected

Then the author clicks the "Upload files" button, both files are uploaded into the system. After that, the students table will show all the uploaded students, as shown in Figure 20 – The students table in the fictional school example.

## Downloading student data into CSV files

All student data in a school grade can be downloaded into a zip file which contains one or two student data CSV files, which contains friends, inclusions and exclusions. The class-allocation tasks and which criteria have been selected in each task with what weights are not included in the downloaded data files.

The uploading and downloading of student data is completely symmetrical. This means that, starting with an empty grade, certain student data CSV files can be uploaded into the system, then all student data in the grade can be downloaded into another set of CSV files, and the uploaded and downloaded CSV files will be identical. It also means that all student data can be downloaded into CSV files, then deleted from the system, then the downloaded data files uploaded back into the system, the data in the system before and after the deletion will be identical.

Teachers are recommended to download the data of their grade after the class allocation is finished, then delete all students from the system. The reasons are

- So that data can be backed up on the school side.



- This minimizes the risk. As stated on the “About Us” page, Teacher's Pet website and database is hosted by Microsoft Azure, where Boeing, Apple, EBay and lots of other global giants host their online solutions, so data is pretty safe. However since uploading the data back into the system takes only a few mouse clicks, it doesn't hurt to be extra cautious.
- Helps us to reduce the hosting price, since we aim at minimizing the fee charged to the schools.

## Excluding a student from a custom/secondary task

Usually all students in the current grade are entered into the system when teachers perform the primary task – allocating classes for next school year. If a student leaves the school he/she should be deleted from the system.

Not all students in the grade always participate in a secondary/custom task such as a camping trip. In this case, do not delete the students that don't participate in this camping trip. Otherwise he/she will disappear from all other tasks, which is not what is intended. Instead, exclude this student from this task. The same student could participate in one task and be excluded from another.

On the “Enter student data” page (see Figure 20 – The students table in the fictional school example), all students are displayed on a table, with each student's data being a row in that table. There is an “Include” column in that table, which is a checkbox. If this checkbox is checked on a student row, it means this student participates in this class-allocation task. All that is needed to exclude a student from this task, is to uncheck this checkbox on that student, then click the floppy disk icon on the column header to save this change:

Students (100)	Email	Current Class	Academic Level	Gender	Secondary Language	Desired Friends	Finalized	Include	Delete	Edit
ALISE COLMENERO	silanliu@gmail.com	1	3	OTHER	CHINESE	JOAQUIN ELKIN, TYREE BRECHT, MELISSIA MCGLOTHLEN, SHANIKA LAMOREAUX, ISA ETHERTON	Orig	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Edit
ARDELLA ALBERTSON	silanstest@gmail.com	1	2	F	CHINESE	HILARIO MCLARTY, BERNIE REGINA, NATHANAEL ENTREKIN, ARTIE FOBBS, DONELLA LAZARD	Orig	<input type="checkbox"/>	<input type="checkbox"/>	Edit
BRAIN BAZEMORE	silanstest2@gmail.com	1	3	M	FRENCH	SANTOS YIP, TREY COPAS, SUNNI STANGL, GINA MINEAR, VELMA MCMILLIAN	Orig	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Edit
CECILY REESE	[test]	1	2	F	CHINESE	TREY COPAS, ARIEL PUCCI, TOWANDA BANGS, GEORGIANA VANCELEAVE, SHANTA OBERMILLER	Orig	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Edit
CHONG FLOURNOY	[test]	1	2	M	FRENCH	EVAN RAPPAPORT, HILARIO MCLARTY, CLIFFORD ELLINGSWORTH, TYREE BRECHT, TERRESA NAU	Orig	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Edit
DANIEL FRISK	[test]	1	2	M	CHINESE	SANTOS YIP, MARGARITO CARBONE, QIANA BOURLAND, AYSIA TURNBULL, PEGGIE COSNER	Orig	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Edit

Figure 43 - How to exclude a student from an event



## Finalizing and unfinalizing data input

After all students have been entered into the system, the last thing that must be done before starting class allocation, is to finalize the data input. It shall be done at the point in time when the teacher says:

*"That's all for this school year. Later in this year, there may be a few students come and go, but the majority of them have been entered into the system."*

To finalize data input, click the "Finalize" link above the students table:

Step 1: Pick the class allocation criteria >> **Step 2: Data Input** >> Step 3: Allocate Classes

Class-Allocation Task: Creating classes for the new school year

Upload Data **Finalize** New Student Exclusions/Inclusions Pick My Friends Download Data

Students (100)	Email	Current Class	Academic Level	Gender	Secondary Language	Desired Friends	Include	Delete	Edit
ALISE COLMENERO	silanliu@gmail.com	1	3	OTHER	CHINESE	JOAQUIN ELKIN, TYREE BRECHT, MELISSIA MCGLOTHLEN, SHANIKA LAMOREAUX, ISA ETHELTON	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Edit
ARDELLA ALBERTSON	silanstest@gmail.com	1	2	F	CHINESE	HILARIO MCLARTY, BERNIE REGINA, NATHANAEAL ENTREKIN, ARTIE FOBBS, DONELLA LAZARD	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Edit
BRAIN BAZEMORE	silanstest2@gmail.com	1	3	M	FRENCH	SANTOS YIP, TREY COPAS, SUNNI STANGL, GINA MINEAR, VELMA MCMILLIAN	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Edit

Figure 44 - The "Finalize" link

After finalization, the page will become

Upload Data **Unfinalize** New Student Exclusions/Inclusions Pick My Friends Download Data

Students (100)	Email	Current Class	Academic Level	Gender	Secondary Language	Desired Friends	Finalized	Include	Delete	Edit
ALISE COLMENERO	silanliu@gmail.com	1	3	OTHER	CHINESE	JOAQUIN ELKIN, TYREE BRECHT, MELISSIA MCGLOTHLEN, SHANIKA LAMOREAUX, ISA ETHELTON	Orig	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Edit
ARDELLA ALBERTSON	silanstest@gmail.com	1	2	F	CHINESE	HILARIO MCLARTY, BERNIE REGINA, NATHANAEAL ENTREKIN, ARTIE FOBBS, DONELLA LAZARD	Orig	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Edit
BRAIN BAZEMORE	silanstest2@gmail.com	1	3	M	FRENCH	SANTOS YIP, TREY COPAS, SUNNI STANGL, GINA MINEAR, VELMA MCMILLIAN	Orig	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Edit
CECILY REESE	[test]	1	2	F	CHINESE	TREY COPAS, ARIEL PUCCI, TOWANDA BANGS, GEORGIANA VANCLEAVE, SHANTA OBERMILLER	Orig	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Edit

Figure 45 - After finalizing

Once finalized, the system will enforce two constraints:



1. New students can still be added, but the number of new students added cannot be more than  $\frac{1}{3}$  of the number of finalized students. In other words, after 100 students are entered and finalized, at most 33 new students can be added.
2. For any class-allocation task such as a camping trip, students enrolled in this task must at least have  $\frac{1}{3}$  being the original students (who were already there when finalized). In other words, if 60 of the 100 students attends a camping trip, among these 60 participants, at least 20 of them must be the original students.

Apart from these two constraints, everything else works the same as before the finalization: changing any student data, downloading students, deleting all of them, uploading them back into the system, etc.

As said earlier, the teachers are supposed to only finalize when they believe they have entered all students in their grade into the system. Having 34 new students transferred into a 100-student grade in the middle of a school year is extremely unlikely to happen.

On the other hand, suppose six new students are transferred into a 100-student grade in the middle of a school year, the teacher will probably never use Teacher's Pet to plan for an event in which only the six new students attend.

Therefore, in reality, these two constraints will have absolutely no impact on the usage of Teacher's Pet. The only time when it does impact, is if a teacher accidentally finalizes the data input when lots of students haven't been entered into the system. In this case the teacher can click the "contact us" link, provide adequate explanation, we will be able to manually unfinalize for him/her.

In Figure 45 - After finalizing there is an extra "Finalized" column in comparison to the student table shown in Figure 20 – The students table in the fictional school example. This extra column tells whether a student was one of the original students when data input was finalized, or added after the finalization. The maximum number of students with this column showing "New" can only be  $\frac{1}{3}$  of the number of students with this column showing "Orig".



In-depth: After finalization, why can the teacher delete all students then add them back? Didn't we just say only 1/3 of new students can be added?

When data input is finalized, the system computes a hash, something like "D8EB549D...5B33EE", from each student's name, and stores them.

When the students are deleted, these hashes are not deleted. This does not defeat the purpose of deleting the students, because even if hackers steal these hashes, they cannot reverse-engineer the student names. It is mathematically impossible. So the hashes contain no meaningful information at all.

When a student is entered into the portal, the system computes a hash from the student name, and compares it with the stored hashes. If it matches one of them, the portal knows that this student was there when data input was finalized, and hence does not count this student as a new student.

When data input is unfinalized, these hashes are deleted.

Once data input is finalized, class allocation can be done as many times as the user wants, in as many tasks as he/she wants, throughout the school year.

Eight months after data input is finalized, the user will be able to unfinalize the data input for this grade him/herself. Before that period elapses, the "Unfinalize" text in Figure 45 - After finalizing is not a clickable link. After that time elapses, it will become a clickable link, which can be clicked to unfinalize the data input for the current grade. The user only needs to ask for us to help he/she wants to unfinalize before the eight-month waiting period elapses.

After unfinalization, the two constraints discussed earlier will be removed.

Typically, teachers would unfinalize their grade when they are about to do the class allocation of the same grade a year later. For example, in term four last year, the year-three teachers allocated their students into new classes for this year. Now it is term four again, the year-three students that were entered into the system last year are now year-four, so they are the year-four teacher's problem now. Now the year-three teachers need to do the class allocation for their current year-three students. In this case, they would unfinalize, delete the old students from year-three, and enter the new year-three students.



## Step 3: Creating the Class Lists

After all data needed by the system have been entered into the system, the teacher can click the “Step 3: Allocate Classes” link to allocate the students into the classes created in step 1. When this link is clicked, a progress bar will show for a second or two, then the “allocate classes” page will show, with a message box on top, displaying a summary of the allocation:

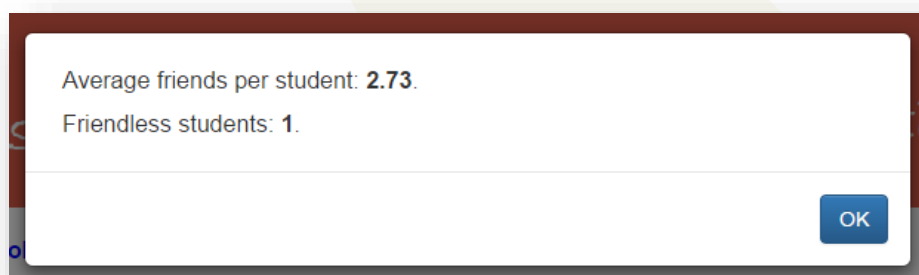


Figure 46 - Class allocation summary message box

Click “OK”, the “Allocate Classes” page is shown:



Step 1: Pick the class allocation criteria >> Step 2: Data Input >> **Step 3: Allocate Classes** **Allocate Classes** [Allocated] 2017-02-06 17:09

**Saved allocation results:**  
<Create New> H

**Classing rule weightings ⓘ :**

**Academic levels mixed** 50  
[Slider bar]

**Even class sizes** 50  
[Slider bar]

**Friends** 50  
[Slider bar]

**At least one friend** 0  
[Slider bar]

**Gender balance** 50  
[Slider bar]

**New students in class** 50  
[Slider bar]

**Secondary language** 50  
[Slider bar]

**Must exclude** 50  
[Slider bar]

4A (25)	4B (25)	4C (25)	4D (25)
Darline Nicholes	Alise Colmenero	Cecily Reese	Ardella Albertson
Eboni Gibbens	Donella Lazard	Daniel Frisk	Brain Bazemore
Elliot Helfrich	Joaquin Elkin	Edyth Aguayo	Chong Flournoy
Elyse Pinner	Shanta Obermiller	Leo Mefford	Lai Fine
Magen Evensen	Tyree Brecht	Leonard Sisco	Lawrence Arguelles
Megan Bowley	Belia Cortese	Wai Tillison	Qiana Bourland
Romona Rufus	Carrol Bealer	Xavier Cheesman	Adriana Mckittrick
Carole Caudill	Darron Poulsen	Darren Levin	Evelia Kinsella
Gina Minear	David Pitter	Georgiana Vancleave	Hilario Mcarty
Kera Arsenaault	Florance Lapeyrouse	Hassan Balke	Leslie Speer
Pat Sowell	Ophelia Letourneau	Lionel Heyen	Maryjane Sippel
Pearlene Hocker	Tamekia Timmins	Rosario Manos	Tomeka Eadie
Stan Claudio	Cheyenne Raver	Danille Franzoni	Artie Fobbs
Ward Ledener	Dalton Greenlee	Ellis Mosteller	Clifford Ellingsworth
Alysia Turnbull	Jarrett Depaoli	Elton Cokley	Del Seckman
Bernie Regina	Kenyetta Bassin	Ena Dangelo	Isa Etherton
Ernesto Sica	Peggie Cosner	Santos Yip	Marceline Strohl
Johnathan Edelson	Ruben Gao	Tanika Seagraves	Shanika Lamoreaux
Troy Laiche	Deena Berti	Zane Gambrel	Velma Momilian
Velva Litchfield	Dorothy Hurwitz	Euna Forward	Arnoldo Melito
Ariel Pucci	Dylan Bien	Francine Merlos	Dexter Welling
Evan Rappaport	Kieth Poehler	Margarito Carbone	Guillermo Gaulin
Frank Proudfoot	Melissia McGlothlen	Sunni Stangl	Jason Rodd
Nathanael Entrekkin	Murray Kulas	Terresa Nau	Morris Dorais
Werner Perreira	Trey Copas	Towanda Bangs	Stephenie Madill

Figure 47. Step 3: allocate classes

## Weightings of the criteria

As shown in the above figure, each class-allocation criteria selected in step 1 (Step 1: Picking the Class Allocation Criteria) is shown here as a slider bar. Slide the slider bars to assign different weightings to the criteria. The very first class allocation was done at the time when this page is loaded, with all weightings being 50. When this page is shown for the first time, it displays the result of that class allocation. To change the weightings, slide the slider bars, then click the "Allocate Classes" button. The engine finishes the allocation following the new set of weightings in a few seconds. The new weightings are automatically saved when the





"Allocate Classes" button is clicked, so that after the teacher shuts down the browser and later logs back in, the latest weightings are loaded automatically.

By nature, most of the class allocation criteria conflict with each other. For example, if the user wants the "friends" criterion to take priority, the "secondary language" criterion will have to give in. It is mathematically impossible to satisfy every criterion to the maximum degree at the same time. The end result is a result of compromise between the conflicting criteria. The weightings of the criteria are how the teacher influences the end result, in determining which criterion is given more priority and which less by the mathematical algorithm in the class allocation process. The teacher can change the weightings, allocate, see the effects, then change weightings and allocate again, and so on. He/she can repeat this process as many times as he/she wants. The system does the heavy lifting, while the teacher has full control over the outcome. It is like a parent drives his/her children to school - it is not a machine that sends them to school, it is the parent, because he/she has full control over the machine.

The absolute values of the weightings in the form of numbers are insignificant; it is how the numbers compare with each other that counts. For example, all weightings being 20 would produce identical class-allocation result as if all weightings are 40. Likewise, if altogether there are three criteria A, B and C, their weightings being 1, 2 and 3 respectively will produce identical result as if they are 10, 20 and 30.

## Verifying the allocation result

On the class allocation page, the effect of the class allocation, i.e. how each criterion has been catered for in the result, can be visualized in an intuitive manner. First click a criterion, then a student, all other students that are associated with this clicked student through the clicked criterion will be highlighted in yellow.

For example, in the fictional school example, if the author clicks the "friends" criterion, then a student, all of his/her friends will be highlighted:



Step 1: Pick the class allocation criteria >> Step 2: Data Input >> **Step 3: Allocate Classes** **Allocate Classes** [Allocated] 2017-02-06 06:09

**Saved allocation results:**  
<Create New>

**Classing rule weightings :**

**Academic levels mixed** 50

**Even class sizes** 50

**Friends** 50

**At least one friend** 0

**Gender balance** 50

**New students in class** 50

**Secondary language** 50

**Must exclude** 50

4A (25)	4B (25)	4C (25)	4D (25)
Darline Nicholes	Alise Colmenero	Cecily Reese	Ardella Albertson
Eboni Gibbens	Donella Lazard	Daniel Frisk	Brain Bazemore
Elliot Helfrich	Joaquin Elkin	Edyth Aguayo	Chong Flournoy
Elyse Pinner	Shanta Obermiller	Leo Mefford	Lai Fine
Magen Evensen	Tyree Brecht	Leonard Sisco	Lawerence Arguelles
Megan Bowley	Belia Cortese	Wai Tillison	Qiana Bourland
Romona Rufus	Carrol Bealer	Xavier Cheesman	Adriana Mokitrick
Carole Caudill	Darron Poulsen	Darren Levin	Evelia Kinsella
Gina Minear	David Pitter	Georgiana Vancleave	Hilario Molarty
Kera Arsenaull	Florance Lapeyrouse	Hassan Balke	Leslie Speer
Pat Sowell	Ophelia Letourneau	Lionel Heyen	Maryjane Sippel
Pearlene Hocker	Tamekia Timmins	Rosario Manos	Tomeka Eadie
Stan Claudio	Cheyenne Raver	Danille Franzoni	Artie Fobbs
Ward Lederer	Dalton Greenlee	Ellis Mosteller	Clifford Ellingsworth
Alysia Turnbull	Jarrett Depaoli	Elton Cokley	Del Seckman
Bernie Regina	Kenyetta Bassin	Ena Dangelo	Isa Etherton
Ernesto Sica	Peggie Cosner	Santos Yip	Marceline Strohl
Johnathan Edelson	Ruben Gao	Tanika Seagraves	Shanika Lamoreaux
Troy Laiche	Deena Berti	Zane Gambrel	Velma Mcmillian
Velva Litchfield	Dorathy Hurwitz	Euna Forward	Arnoldo Melito
Ariel Pucci	Dylan Bien	Francine Merlos	Dexter Welling
Evan Rappaport	Kieth Poehler	Margarito Carbone	Guillermo Gaulin
Frank Proudfoot	Melissia Moglothlen	Sunni Stangl	Jason Rodd
Nathanael Entrekin	Murray Kulas	Terresa Nau	Morris Dorais
Werner Ferreira	Trey Copas	Towanda Bangs	Stephenie Madill

Figure 48 - Verifying class allocation result

The pink student is the one clicked, the yellow students are his desired friends.

If the author clicks the gender criterion, then a student, all students of the same gender as the clicked student will be highlighted, and the tooltip tells what gender are these highlighted students:

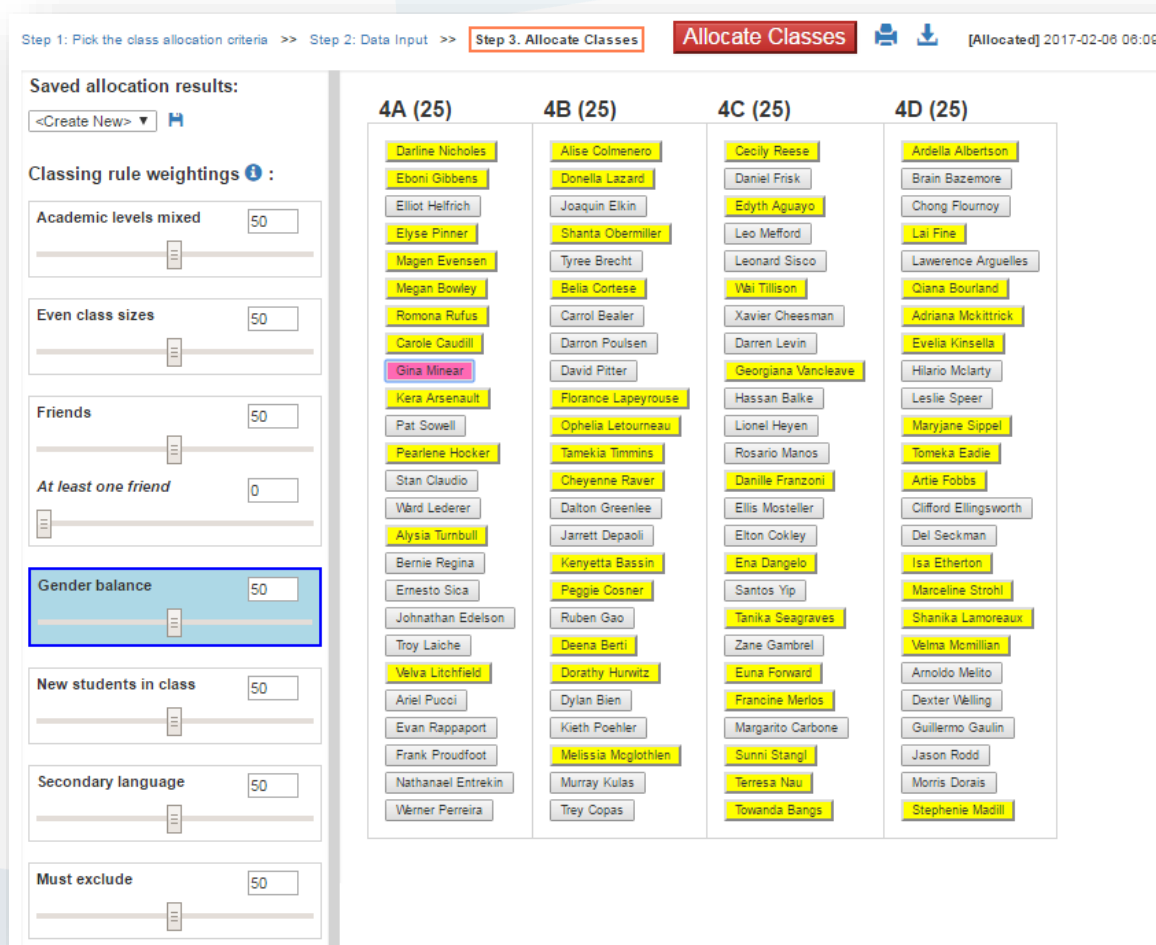


Figure 49 - Verifying class allocation result (2)

## Verifying the intelligence of the algorithm



Now that the allocation result is the compromise of many criteria, it is most likely that none of the criteria is fully satisfied in the allocation result. How does the user know for sure that the algorithm is really working?


To gain some insight on the intelligent algorithm, start with the first criterion. Set its weighting to a non-zero value, and set all other criteria's weightings to zero. Do an allocation. The result will fully satisfy this criteria. Then, set this criterion to zero, and set the next criterion to non-zero. The result will fully satisfy this criterion. Test each criterion in this manner. Each criterion is fully satisfied when there is no other criterion to compete with it.




Then, more sophisticated testing can be done by turning on more than one criteria at the same time. One of the YouTube demo videos on the home page demonstrates such a test, using the test data for year 3 in the fictional school.


First, turn on only the “secondary language” criterion, which demands students studying the same secondary language to be in the same class. Do an allocation. The following will be shown:


Step 1: Pick the class allocation criteria >> Step 2: Data Input >> **Step 3: Allocate Classes** **Allocate Classes**  


**Saved allocation results:**  
<Create New> 


**Classing rule weightings ⓘ :**


Academic levels mixed   



Even class sizes   



Friends   


At least one friend   


Gender balance   


New students in class   


**Secondary language**   


Must exclude   


4A (53)	4B (47)	4C (0)	4D (0)
Brain Bazemore			
Chong Flournoy			
Donella Lazard			
Eboni Gibbens	Alise Colmenero		
Elliot Helfrich	Ardella Albertson		
Joaquin Elkin	Cecily Reese		
Lawrence Arguelles	Daniel Frisk		
Leo Mefford	Darline Nicholes		
Megan Bowley	Edyth Aguayo		
Qiana	Secondary Language = FRENCH		
Romona Rufus	Lai Fine		
Shanta Obermiller	Leonard Sisco		
Tyree Brecht	Magen Evensen		
Xavier Cheesman	Wai Tillison		
Adriana Mckittrick	Carrol Bealer		
Bella Cortese	Darren Levin		
Carole Caudill	David Pitter		
Darron Poulsen	Evelia Kinsella		
Florance Lapeyrouse	Georgiana Vancleave		
Gina Minear	Leslie Speer		
Hassan Balke	Lionel Heyen		
Hilario Molarty	Pearlene Hooker		
Kera Arsenault	Rosario Manos		
Maryjane Sippel	Stan Claudio		
Ophelia Letourneau	Tamekia Timmins		
Pat Sowell	Artie Fobbs		
Tomeka Eadie	Bernie Regina		
Ward Lederer	Clifford Ellingsworth		
Alysia Turnbull	Dalton Greenlee		
Cheyenne Raver	Danille Franzoni		
Pat Sowell	Ethan Cokley		

Figure 50 – Algorithm testing step 1



The algorithm allocated all French-studying students into class 4A, and all Chinese-studying students into 4B, leaving the other two classes empty. This, obviously satisfies the “secondary language” criterion completely.

Now, increase the “even class size” criterion to 50, and allocate. The following dialog is shown:

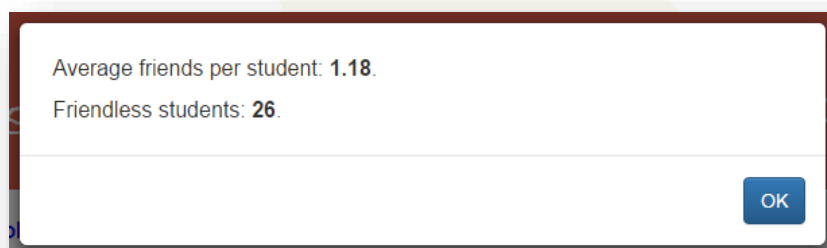


Figure 51 – Algorithm testing step 2 – 1

Click “OK”, the following is shown:



Step 1: Pick the class allocation criteria >> Step 2: Data Input >> **Step 3: Allocate Classes** **Allocate Classes**

**Saved allocation results:**  
<Create New>

**Classing rule weightings :**

**Academic levels mixed**

**Even class sizes**

**Friends**

**At least one friend**

**Gender balance**

**New students in class**

**Secondary language**

**Must exclude**

4A (25)	4B (25)	4C (25)	4D (25)
Alysia Turnbull	Alise Colmenero	Ardella Albertson	Brain Bazemore
Cheyenne Raver	Cecily Reese	Darline Nicholes	Chong Flournoy
Del Seckman	Daniel Frisk	Lai Fine	Donella Lazard
Ellis Mosteller	Edyth Aguayo	Wai Tillison	Eboni Gibbens
Ernesto Sica	Elyse Pinner	David Pitter	Elliot Helfrich
Isa Etherton	Leonard Sisco	Hilario Mcclarty	Joaquin Elkin
Johnathan Edelson	Magen Evensen	Leslie Speer	Lawrence Arguelles
Marceline Strohl	Carrol Bealer	Ophelia Letourneau	Leo Mefford
Peggie Cosner	Darren Levin	Rosario Manos	Megan Bowley
Ruben Gao	Evella Kinsella	Ward Lederer	Qiana Bourland
Troy Laiche	Georgiana Vancleave	Artie Fobbs	Romona Rufus
Zane	Pearlene Hocker	Dalton Greenlee	Shanta Obermiller
Arnoldo Melito	Stan Claudio	Ena Dangelo	Tyree Brecht
Deena Berti	Tamekia Timmins	Jarrett Depaoli	Xavier Cheesman
Dexter Welling	Bernie Regina	Santos Yip	Adriana Mckittrick
Euna Forward	Clifford Ellingsworth	Shanika Lamoreaux	Bela Cortese
Evan Rappaport	Danille Franzoni	Velma Mcmillan	Carole Caudill
Francine Merlos	Elton Cokley	Velva Litchfield	Darron Poulsen
Guillermo Gaulin	Kenyetta Bassin	Dorothy Hurwitz	Florance Lapeyrouse
Melissia Mclothlen	Tanika Seagraves	Dylan Bien	Gina Minear
Morris Dorais	Ariel Pucci	Jason Rodd	Hassan Baika
Stephenie Madill	Frank Proudfoot	Kieth Poehler	Kera Arsenault
Teresa Nau	Margarito Carbone	Murray Kulas	Maryjane Sippel
Trey Copas	Sunni Stangl	Nathanael Entrekin	Pat Sowell
Verner Perreira		Towanda Bangs	Tomeka Eadie

Secondary Language = FRENCH

Figure 52 - Algorithm testing step 2 - 2



To satisfy the “even class size” criterion, the algorithm has to make all four classes the same size. So it is forced to move a few French-studying students into a Chinese-studying class, but it did its best to keep the “secondary language” criterion satisfied. In average each student has 1.18 friends, and there are 26 friendless students. This is because the “friends” criterion’s weighting is zero, so the algorithm totally disregarded it. Next, increase the “friends” weighting to 12, and allocate. The following result is shown:




Average friends per student: 2.01.  
Friendless students: 10.


OK


Figure 53 - Algorithm testing step 3 – 1


Step 1: Pick the class allocation criteria >> Step 2: Data Input >> **Step 3. Allocate Classes** **Allocate Classes**  


**Saved allocation results:**  
<Create New> 


**Classing rule weightings ⓘ :**


**Academic levels mixed** 0  



**Even class sizes** 50  



**Friends** 12  


**At least one friend** 0  


**Gender balance** 0  


**New students in class** 0  


**Secondary language** 50  


**Must exclude** 0  


4A (25)	4B (25)	4C (25)	4D (25)
Donella Lazard	Brain Bazemore	Alise Colmenero	Ardella Albertson
Joaquin Elkin	Chong Flournoy	Cecily Reese	Daniel Frisk
Leo Mefford	Eboni Gibbens	Leonard Sisco	Darline Nicholes
Megan Bowley	Elliot Helfrich	Carrol Bealer	Edyth Aguayo
Romona Rufus	Lawerence Arguelles	Darren Levin	Elyse Pinner
Xavier Cheesman	Qiana Bourland	David Pitter	Lai Fine
Adriana McKittrick	Shanta Obermiller	Evelia Kinsella	Magen Evensen
Carole Caudill	Tyree Brecht	Georgiana Vancleave	Wai Tillison
Darron Poulsen	Bella Cortese	Gina Minear	Leslie Speer
Kera Arsenaunt	Florance Lapeyrrouse	Pearlene Hocker	Lionel Heyen
Ophelia Letourneau	Hassan Balke	Rosario Manos	Stan Claudio
Tomeka Eadie	Hilario Molarty	Tamekia Timmins	Artie Fobbs
Ward Lederer	Marylane Siobel	Dalton Greenlee	Bernie Regina
Cheyenne	Secondary Language = FRENCH	Ena Dangelo	Clifford Ellingsworth
Ellis Mosteller	Alysia Turnbull	Jarrett Depaoli	Danille Franzoni
Isa Etherton	Del Seckman	Ruben Gao	Elton Cokley
Johnathan Edelson	Ernesto Sica	Shanika Lamoreaux	Kenyetta Bassin
Marceline Strohl	Peggie Cosner	Tanika Seagraves	Santos Yip
Zane Gambrel	Troy Laiche	Velva Litchfield	Velma Mcmillian
Dexter Welling	Arnoldo Melito	Jason Rodd	Ariel Pucci
Euna Forward	Deena Berti	Margarito Carbone	Dorathy Hurwitz
Francine Merlos	Evan Rappaport	Melissia Moglothien	Dylan Bien
Guillermo Gaulin	Morris Dorais	Murray Kulas	Frank Proudfoot
Stephenie Madill	Terresa Nau	Sunni Stangl	Kieth Poehler
Werner Perreira	Trey Copas	Towanda Bangs	Nathanael Entrekin

Figure 54 - Algorithm testing step 3 – 2



The class size remains even, and the “secondary language” criterion hasn’t deteriorated at all, but the “friends” criterion is significantly improved – the average friends per student increased from 1.18 to 2.01, and the friendless students dropped from 26 to 10.

How did the algorithm improve the “friends” criterion without hurting the other two? It worked out that it can swap students between the two French classes to give more friends to students, without affecting the other two criteria.

At last, increase the “friends” weighting to 20. The following result will be shown:

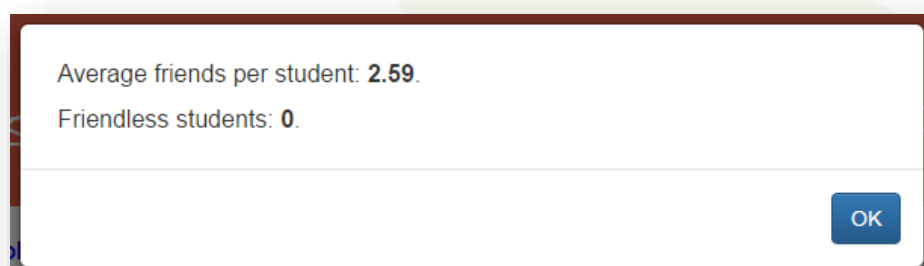





Figure 55 - Algorithm testing step 4 - 1








Step 1: Pick the class allocation criteria >> Step 2: Data Input >> **Step 3: Allocate Classes** **Allocate Classes**  


**Saved allocation results:**  
<Create New> 


**Classing rule weightings ⓘ :**


Academic levels mixed   



Even class sizes   



Friends   


At least one friend   


Gender balance   


New students in class   


**Secondary language**   


Must exclude   


4A (25)	4B (25)	4C (25)	4D (25)
Joaquin Elkin	Ardella Albertson	Alise Colmenero	Cecily Reese
Lawrence Arguelles	Darline Nicholes	Brain Bazemore	Chong Flournoy
Leo Mefford	Donella Lazard	Edyth Aguayo	Daniel Frisk
Megan Bowley	Eboni Gibbens	Lai Fine	Elliot Helfrich
Romona Rufus	Elyse Pinner	Tyree Brecht	Leonard Sisco
Xavier Cheesman	Magen Evensen	Carrol Bealer	Qiana Bourland
Adrian Caudill	Bela Cortese	Rosario Manos	Wai Tillison
Carole Caudill	Bella Cortese	Tomeka Eadie	Darren Levin
Kera Arsenault	Darron Poulsen	Artie Fobbs	Florance Lapeyrouse
Ward Lederer	David Pitter	Clifford Ellingsworth	Georgiana Vancleave
Cheyenne Raver	Evelia Kinsella	Danille Franzoni	Hassan Balke
Ernesto Sica	Gina Minear	Ena Dangelo	Leslie Speer
Johnathan Edelson	Hilario Molarty	Isa Etherton	Lionel Heyen
Marceline Strohl	Maryjane Sippel	Ruben Gao	Ophelia Letourneau
Zane Gambrel	Pat Sowell	Shanika Lamoreaux	Pearlene Hocker
Arnoldo Melito	Stan Claudio	Velma Momillian	Alysia Turnbull
Deena Berti	Tamekia Timmins	Ariel Pucci	Del Seckman
Dexter Welling	Bernie Regina	Dorathy Hurwitz	Ellis Mosteller
Euna Forward	Dalton Greenlee	Frank Proudfoot	Elton Cokley
Francine Merlos	Jarrett Depaoli	Jason Rodd	Santos Yip
Guillermo Gaulin	Kenyetta Bassin	Margarito Carbone	Tanika Seagraves
Morris Dorais	Peggie Cosner	Melissia Moglothlen	Troy Laiche
Stephenie Madill	Velva Litchfield	Murray Kulas	Dylan Bien
Teresa Nau	Kieth Poehler	Sunni Stangl	Evan Rappaport
Werner Perreira	Nathanael Entrekin	Trey Copas	Towanda Bangs

Figure 56 - Algorithm testing step 4 – 2

The “friends” criterion is further improved – the average friends increased from 2.01 to 2.59, and friendless student dropped from 10 to zero, but the “secondary language” criterion starts to suffer.

This test proves how intelligent the algorithm is, how it tries to satisfy all criteria as much as possible, and how the weightings assigned to different criteria controls the priorities of each criterion, when they conflict with each other during the allocation process.



## Number of classes vs. number of friends

The more classes to allocate the students into, the less friends each student gets in his/her new class.

In the figure below, each student is represented by a circle, and each friend request is presented by a red line:

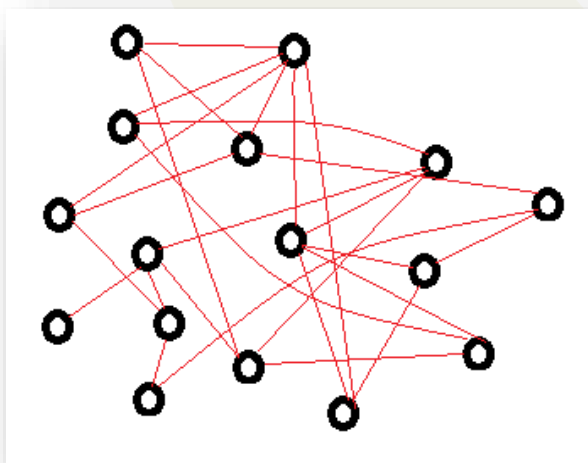


Figure 57 - Friend requests forms a web

The friend requests form a web. If there are only 25 students in one class in the grade, then there is no severing on any lines. This means every student gets all of his/her five desired friends.

If the students' needs to be divided into two classes, then a cut needs to be made across the web somewhere, and some lines will have to be severed, which means friends separated:

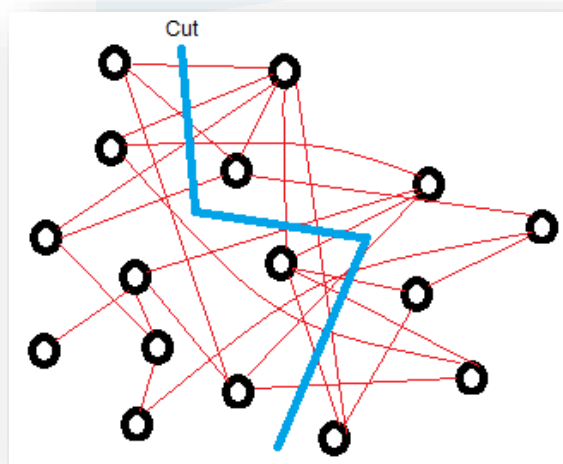


Figure 58 - One cut through the friends web

The more classes the student are divided into, the more cuts will have to be made across this web. If 200 students need to be divided into eight classes, seven cuts will have to be made across the web. If the classes are allocated randomly, the result will be an average less than one friend per student, with half of the students left friendless. With this many students, no matter how hard a teacher tries, the allocation result won't be much better than doing it randomly, even if he/she goes through every single student and check every friend this student wants, because when the teacher moves John to Peter's class to give him a friend, he/she may have deprived a friend from Josh cause John is also Josh's friend.

The more classes to allocate, the bigger difference Teacher's Pet's ingenious mathematical algorithm can make. In the case of dividing 200 students into eight classes, Teacher's Pet can achieve two to three friends per student, with almost no one left friendless, meanwhile catering for five or eight other class-allocation criteria that the teachers want to enforce.

## Sympathy toward friendless students

On the step 3 page (Figure 47. Step 3: allocate classes), the reader may have noticed that the "friends" criterion is different from other criteria in that it has two slider bars. The first slider bar sets the weighing of the "friends" criterion, in the same way as all other criteria, while the second slider bar, which is called "At least one friend", sets the weighting of the first friend of each student. If this second weighting is set to be higher than the first, it means that the user wants the engine to go extra length to guarantee that each student has at least one friend.



This feature is called “Sympathy Toward Friendless Students”. It is very useful in minimizing or even eliminating friendless students.

Consider the following example:

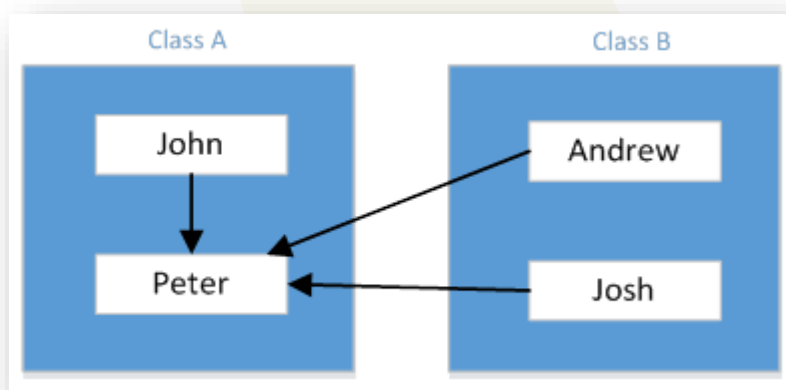


Figure 59 - Example of sympathy toward friendless student

John, Andrew and Josh all want Peter to be in their own classes. John has no other friend in his class except for Peter, while Andrew and Josh have already some friends. If you set the “at least one friend” weighting to lower than the first slider bar, then the engine will move Peter to class B, because this move gives an extra friend to two students, while only one student loses a friend. This would result in John having no friend in his class. If you believe that a student getting his/her first friend is more important than a student getting his/her second or third friend, then you can assign a weighting to the “at least one friend” slider bar that is higher than the first slider bar. This will prevent the above-mentioned move from happening.

It is also useful when the teacher feels that the “friends” criterion suppresses other criteria too much, but, if the “friends” criterion weighting is lowered, there are too many friendless students. In this case, the teacher can lower the “friends” criterion weighting, and increase the “at least one friend” weighting. The effects of this are:

1. If a student has no friend, the engine suppresses the competition from other criteria, including competition for friends from other students who already have at least one friend, so that this student gets his/her first friend.
2. As long as each student has one friend, the engine suppresses the “friends” criterion to not to compete with other criteria.



The effectiveness of this “sympathy toward friendless students” feature can be demonstrated using the test data in the fictional school example.

First, set the weighting of criterion “even class sizes” and “secondary language” to 50, and all other criteria to zero. Keep the “at least one friend” slider bar zero. Gradually increase the weighting of the “friends” criterion from zero. The lowest “friends” weighting that can achieve zero friendless student is 15:

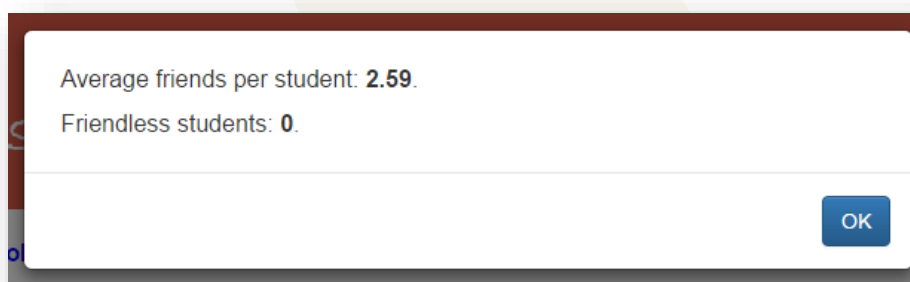


Figure 60 - Testing the sympathy toward friendless student feature – 1

The “secondary language” criterion has taken quite a bit of hit because of the competition from the “friends” criterion:



Step 1: Pick the class allocation criteria >> Step 2: Data Input >> **Step 3: Allocate Classes** **Allocate Classes**

**Saved allocation results:**  
<Create New>

**Classing rule weightings ⓘ :**

**Academic levels mixed** 0

**Even class sizes** 50

**Friends** 15

**At least one friend** 0

**Gender balance** 0

**New students in class** 0

**Secondary language** 50

**Must exclude** 0

4A (25)	4B (25)	4C (25)	4D (25)
Donella Lazard	Daniel Frisk	Alise Colmenero	Brain Bazemore
Joaquin Elkin	Eboni Gibbens	Ardella Albertson	Chong Flournoy
Lawrence Arguelles	Edyth Aguayo	Cecily Reese	Darline Nicholes
Megan Bowley	Elliott Helfrich	Lai Fine	Elyse Pinner
Qiana Bourland	Leo Mefford	Bela Cortese	Leonard Sisco
Adriana Mckittrick	Romona Rufus	Carrol Bealer	Magen Evensen
Carole Caudill	Shanta Obermiller	David Pitter	Tyree Brecht
Darron Poulsen	Gina Minear	Evella Kinsella	Wai Tillison
Kera Arsenault	Hassan Balke	Georgiana Vanleave	Xavier Cheesman
Ophelia Letourneau	Lionel Heyen	Hilario Molarty	Darren Levin
Tomeka Eadie	Maryjane Sippel	Rosario Manos	Florance Lapeyrouse
Ward Lederer	Pearlene Hocker	Artie Fobbs	Leslie Speer
Del Seckman	Tamekia Timmins	Bernie Regina	Pat Sowell
Isa Etherton	Alvina Turnbull	Clifford Ellingsworth	Stan Claudio
Johnathan		Ena Dangelo	Danille Franzoni
Marceline Strohl	Dalton Greenlee	Ruben Gao	Ellis Mosteller
Zane Gambrel	Elton Cokley	Shanika Lamoreaux	Tanika Seagraves
Arnoldo Melito	Ernesto Sica	Ariel Pucci	Velma Mcmillian
Deena Berti	Jarrett Depaoli	Dorothy Hurwitz	Dylan Bien
Dexter Welling	Kenyetta Bassin	Jason Rodd	Evan Rappaport
Euna Forward	Peggie Cosner	Margarito Carbone	Frank Proudfoot
Francine Merlos	Santos Yip	Melissia Mcglothlen	Morris Dorais
Guillermo Gaulin	Troy Laiche	Murray Kulas	Sunni Stangl
Stephenie Madill	Velva Litchfield	Nathanael Entrekin	Terresa Nau
Werner Perreira	Kieth Poehler	Towanda Bangs	Trey Copas

Secondary Language = FRENCH

Figure 61 - Testing the sympathy toward friendless student feature – 2

Pay attention to these two numbers: in average each student has 2.59 friends, and there are 15 French-studying students that are forced into a Chinese-studying class.



Now, change the “friends” weighting to zero, and the “at least one friend” slider bar to 70, and reallocate:




Average friends per student: 1.96.  
Friendless students: 0.


OK


Figure 62 - Testing the sympathy toward friendless student feature – 3

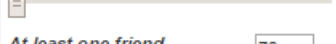
Step 1: Pick the class allocation criteria >> Step 2: Data Input >> **Step 3. Allocate Classes** **Allocate Classes**  


**Saved allocation results:**  
<Create New> 


**Classing rule weightings ⓘ :**

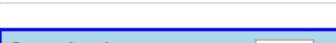
**Academic levels mixed** 0  



**Even class sizes** 50  



**Friends** 0  


**At least one friend** 70  


**Gender balance** 0  


**New students in class** 0  


**Secondary language** 50  


**Must exclude** 0  


4A (25)	4B (25)	4C (25)	4D (25)
Megan Bowley	Alise Colmenero	Ardella Albertson	Daniel Frisk
Adriana Mckittrick	Brain Bazemore	Cecily Reese	Darline Nicholes
Carole Caudill	Chong Flournoy	Eboni Gibbens	Edyth Aguayo
Darron Poulsen	Donella Lazard	Elyse Pinner	Leo Mefford
Ophelia Letourneau	Elliot Helfrich	Joaquin Elkin	Leonard Sisco
Tomeka Eadie	Lawrence Arguelles	Lai Fine	Xavier Cheesman
Cheyenne Raver	Romona Rufus	Magen Evensen	Darren Levin
Johnathan Edelson	Tyree Brecht	Qiana Bourland	Georgiana Vancleave
Marceline Strohl	Wai Tillison	Shanta Obermiller	Lionel Heyen
Ruben Gao	Belia Cortese	Carrol Bealer	Maryjane Sippel
Troy Laiche	Evella Kinsella	David Pitter	Rosario Manos
Zane Gambrel	Florance Lapeyrouse	Leslie Speer	Tamekia Timmins
Arnoldo Melito	Gina Minear	Stan Claudio	Elton Cokley
Deena Berti	Hassan Balke	Artie Fobbs	Jarrett Depaoli
Dexter Welling	Hilario McLarty	Bernie Regina	Kenyetta Bassin
Euna Forward	Kera Arsenaull	Dalton Greenlee	Peggie Cosner
Evan Rappaport	Pat Sowell	Danille Franzoni	Santos Yip
Francine Merlos	Pearlene Hocker	Del Seckman	Shanika Lamoreaux
Guillermo Gaulin	Ward Lederer	Isa Etherton	Tanika Seagraves
Melissia Mcglothlen	Alysia Turnbull	Velma Momillian	Ariel Pucci
Morris Dorais	Clifford Ellingsworth	Dorothy Hurwitz	Frank Proudfoot
Stephenie Madill	Ellis Mosteller	Dylan Bien	Margarito Carbone
Terresa Nau	Ena Dangelo	Jason Rodd	Murray Kulas
Trey Copas	Ernesto Sica	Kieth Poehler	Sunni Stangl
Werner Perreira	Velva Litchfield	Nathanael Entrekin	Towanda Bangs

Figure 63 - Testing the sympathy toward friendless student feature – 4



The average friends per student dropped from 2.59 to 1.96, and the number of French-studying students that were forced into the Chinese-studying class dropped from 15 to 10.

Here is the explanation:

In the first case, where the “friends” weighting is 15, the algorithm was told to disregard how many friends a student already has. Even if a student has already one or two or even more friends, the algorithm will still try to give him/her more friends, at the cost of other criteria, including the “secondary language” criterion.



In the second case, where the “friends” weighting is zero and the “at least one friend” slider bar is 70, the algorithm tried as hard as before to give the first friend to every student. But as soon as a student has one friend, the algorithm no longer tried to give him/her more. The outcome is, of course, the average friends each student dropped significantly, and the “friends” criterion’s competition with other criteria dropped significantly, so the “secondary language” criteria became much better satisfied.


## Dragging & dropping students


Teachers can control the class-allocation outcome not only through the weightings, but also through dragging and dropping the students between the allocated classes. In the year three of the fictional school, after moving three students from class 4A to class 4B, the allocation result becomes the following:








Step 1: Pick the class allocation criteria >> Step 2: Data Input >> **Step 3: Allocate Classes** **Allocate Classes**  


**Saved allocation results:**  
<Create New> 


**Classing rule weightings  :**


**Academic levels mixed**   



**Even class sizes**   



**Friends**   


**At least one friend**   


**Gender balance**   


**New students in class**   


**Secondary language**   


**Must exclude**   




4A (22)	4B (28)	4C (25)	4D (25)
	Ardella Albertson		Cecily Reese
	Darline Nicholes		Chong Flournoy
	Donella Lazard	Alise Colmenero	Daniel Frisk
	Eboni Gibbens	Brain Bazemore	Elliot Helfrich
	Elyse Pinner	Edyth Aguayo	Leonard Sisco
	Leo Mefford	Lai Fine	Qiana Bourland
	Magen Evensen	Tyree Brecht	Wai Tillison
	Megan Bowley	Carrol Bealer	Darren Levin
	Romona Rufus	Rosario Manos	Florance Lapeyrouse
	Shanta Obermiller	Tomeka Eadie	Georgiana Vancleave
	Belia Cortese	Artie Fobbs	Hassan Balke
	Darron Poulsen	Clifford Ellingsworth	Leslie Speer
	David Pitter	Danille Franzoni	Lionel Heyen
	Evellia Kinsella	Ena Dangelo	Ophelia Letourneau
	Gina Minear	Isa Etherton	Pearlene Hooker
	Hilario McLarty	Ruben Gao	Alysia Turnbull
	Maryjane Sippel	Shanika Lamoreaux	Del Seckman
	Pat Sowell	Velma Mcmillian	Ellis Mosteller
	Stan Claudio	Ariel Pucci	Elton Cokley
	Tamekia Timmins	Dorothy Hurwitz	Santos Yip
	Bernie Regina	Frank Proudfoot	Tanika Seagraves
	Dalton Greenlee	Jason Rodd	Troy Laiche
	Jarrett Depaoli	Margarito Carbone	Dylan Bien
	Kenyetta Bassin	Melissia Mcglothlen	Evan Rappaport
	Peggie Cosner	Murray Kulas	Towanda Bangs
	Velva Litchfield	Sunni Stangl	
	Kieth Poehler	Trey Copas	
	Nathanael Entrekin		
Joaquin Elkin			
Lawrence Arguelles			
Xavier Cheesman			
Adriana McKittrick			
Carole Caudill			
Kera Arsenaunt			
Ward Lederer			
Cheyenne Raver			
Ernesto Sica			
Johnathan Edelson			
Marceline Strohl			
Zane Gambrel			
Arnoldo Melito			
Deena Berti			
Dexter Welling			
Euna Forward			
Francine Merlos			
Guillermo Gaulin			
Morris Dorais			
Stephenie Madill			
Terresa Nau			
Werner Perreira			


Figure 64 - Class allocation result after drag and drop

The three moved students are marked with red dashed border.


Click such a moved student, the clicked student becomes pink, and the original class from which this student was moved becomes red dashed bordered:




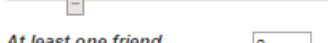
Step 1: Pick the class allocation criteria >> Step 2: Data Input >> **Step 3: Allocate Classes** **Allocate Classes**  


**Saved allocation results:**  
<Create New> 

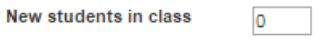
**Classing rule weightings ⓘ :**


**Academic levels mixed** 0  



**Even class sizes** 50  


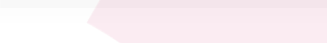
**Friends** 20  


**At least one friend** 0  


**Gender balance** 0  


**New students in class** 0  


**Secondary language** 50  


**Must exclude** 0  


4A (22)	4B (28)	4C (25)	4D (25)
Joaquin Elkin	Ardella Albertson	Alise Colmenero	Cecily Reese
Lawrence Arguelles	Darline Nicholes	Brain Bazemore	Chong Flournoy
Xavier Cheesman	Donella Lazard	Edyth Aguayo	Daniel Frisk
Adriana Mckittrick	Eboni Gibbens	Lai Fine	Elliot Helfrich
Carole Caudill	Elyse Pinner	Tyree Brecht	Leonard Sisco
Kera Arsenaunt	Leo Mefford	Carrol Bealer	Qiana Bourland
Ward Lederer	Magen Evensen	Rosario Manos	Wai Tillison
Cheyenne Raver	Megan Bowley	Tomeka Eadie	Darren Levin
Ernesto Sica	Romona Rufus	Artie Fobbs	Florance Lapeyrouse
Johnathan Edelson	Shanta Obermiller	Clifford Ellingsworth	Georgiana Vancleave
Marceline Strohl	Bella Cortese	Danille Franzoni	Hassan Balke
Zane Gambrel	Darron Poulsen	Ena Dangelo	Leslie Speer
Arnoldo Melito	David Pitter	Isa Etherton	Lionel Heyen
Deena Berti	Evelia Kinsella	Ruben Gao	Ophelia Letourneau
Dexter Welling	Gina Minear	Shanika Lamoreaux	Pearlene Hocker
Euna Forward	Hilario Mclarty	Velma Mcmillian	Alysia Turnbull
Francine Merlos	Maryjane Sippel	Ariel Pucci	Del Seckman
Guillermo Gaulin	Pat Sowell	Dorathy Hurwitz	Ellis Mosteller
Morris Dorais	Stan Claudio	Frank Proudfoot	Elton Cokley
Stephanie Madill	Tamekia Timmins	Jason Rodd	Santos Yip
Teresa Nau	Bernie Regina	Margarito Carbone	Tanika Seagraves
Werner Perreira	Dalton Greenlee	Melissia Mcglothlen	Troy Laiche
	Jarrett Depaoli	Murray Kulas	Dylan Bien
	Kenyetta Bassin	Sunni Stangl	Evan Rappaport
	Peggie Cosner	Trey Copas	Towanda Bangs
	Velva Litchfield		
	Kieth Poehler		
	Nathanael Entrekin		

Figure 65 - Showing which class a student was dragged from

Such drag and drops are automatically saved, so that next time the user logs in, the drag and drops are still there.

If "Allocate Classes" button is clicked, following warning will be shown:

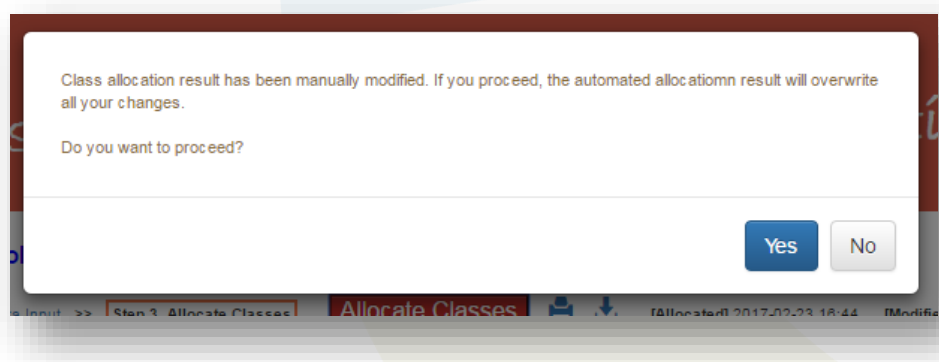


Figure 66 - Warning about losing drag and drops

If “Yes” is clicked, the classes will be reallocated, and all previous drag-and-drops will be lost.

## Saving allocation results

Sometimes, after many iterations of toiling with the weightings, the teacher finally has a satisfactory class-allocation result. He/she still wants to try further to see if an even better outcome can be achieved, but he/she wants to save the current satisfactory outcome, so that, if further tries fail to have a better outcome, he/she can fall back to this saved result.

To save the current allocation result, just click the floppy disk icon above the class-allocation criteria slider bars:



Step 1: Pick the class allocation criteria >> Step 2: Data Input >> **Step 3. Allocate Class**

**Saved allocation results:**

<Create New> ▼

**Classing rule weightings ⓘ :**

Academic levels mixed 0

Even class sizes 50

Friends 20

At least one friend 0

**4A (25)**

Joaquin Elkin  
Lawrence Arguelles  
Leo Mefford  
Megan Bowley  
Romona Rufus  
Xavier Cheesman  
Adriana McKittrick  
Carole Caudill  
Kera Arsenaull  
Ward Lederer  
Cheyenne Raver  
Ernesto Sica  
Johnathan Edelson  
Marceline Strohl

**4B (2)**

Ardella  
Darline  
Donell  
Eboni  
Elyse  
Magen  
Shanta  
Bella C  
Darron  
David  
Evelia  
Gina M  
Hilario  
Maryja

Figure 67 - When no allocation result is saved

A dialog will be popped up to ask the user to provide a name for this saved result:

Please enter an unique name for the class allocation result (less than 20 characters):

My first

**Save Class Allocation Result** **Cancel**



Figure 68 - Entering a name for a saved allocation result


Enter a name shorter than 20 characters, then click the “Save Class Allocation Result” button, the result is saved, and the name of the saved result will appear in the dropdown box:



Step 1: Pick the class allocation criteria >> Step 2: Data Input >> **Step 3: Allocate Classes**

**Saved allocation results:**

<Create New>   
My first 

**Classing rule weightings ** :

Academic levels mixed 0

Even class sizes 50

Friends 20

At least one friend 0

**4A (25)**

Joaquin Elkin	Ardella Albertson
Lawrence Arguelles	Darline Nicholes
Leo Mefford	Donella Lazard
Megan Bowley	Eboni Gibbens
Romona Rufus	Elyse Pinner
Xavier Cheesman	Magen Evensen
Adriana Mokittrick	Shanta Obermiller
Carole Caudill	Bella Cortese
Kera Arsenaull	Darron Poulsen
Ward Lederer	David Pitter
Cheyenne Raver	Evellia Kinsella
Ernesto Sica	Gina Minear
Johnathan Edelson	Hilario McLarty
Marceline Strohl	Maryjane Sippel
Zane Gambrel	Pat Sowell
Arnoldo Melito	Stan Claudio

**4B (25)**




Ardella Albertson
Darline Nicholes
Donella Lazard
Eboni Gibbens
Elyse Pinner
Magen Evensen
Shanta Obermiller
Bella Cortese
Darron Poulsen
David Pitter
Evellia Kinsella
Gina Minear
Hilario McLarty
Maryjane Sippel
Pat Sowell
Stan Claudio


Figure 69 - Saved allocation result

To revert to or delete a previously saved allocation result, select that result on the dropdown box. Two more icons will appear to the right of the floppy disk icon:

Step 1: Pick the class allocation criteria >> Step 2: Data Input >> **Step 3: Allocate Classes**

**Saved allocation results:**

My first   

**Classing rule weightings ** :

Academic levels mixed 0

Even class sizes 50

**4A (25)**

Chong Flournoy
Lawrence Arguelles
Leo Mefford
Megan Bowley
Qiana Bourland
Romona Rufus
Adriana Mckittrick
Carole Caudill
Hilario McLarty

**4B (25)**

Cecily Reese
Daniel Frisk
Darline Nicholes
Edyth Aguayo
Lai Fine
Magen Evensen
Shanta Obermiller
Xavier Cheesman
Georgiana Vanc

Figure 70 - When a saved allocation result is selected

Click the “refresh” icon on the left to load this previous allocation result, or the cross icon on the right to delete it.



# Appendixes

## Sample student data

Just copy the text inside the grey box, paste it into a text editor, and save the text file as the given file name.

NOTE: when copying text from a PDF document that spans multiple pages, the text in the page header and footer, such as "PAGE 57 of 60", is copied too. When pasting the copied text it into the text file, please carefully remove the header and footer. Otherwise the upload process will fail.

First file is named "Year 3 - Creating classes for the new school year - student data.csv", with the following content:

```
First Name,Middle Name>Last Name,Current Class,Email,Academic Level,Gender,Secondary Language,Desired Friends
ALISE,,COLMENERO,1,silanliu@gmail.com,3,F,Chinese,"JOAQUIN ELKIN, TYREE BRECHT, MELISSIA MCGLOTHLEN, SHANIKA LAMOREAUX, ISA ETHERTON"
ARDELLA,,ALBERTSON,1,silanstest@gmail.com,2,F,Chinese,"HILARIO MCLARTY, BERNIE REGINA, NATHANAEL ENTREKIN, ARTIE FOBBS, DONELLA LAZARD"
BRAIN,,BAZEMORE,1,silanstest2@gmail.com,3,M,French,"SANTOS YIP, TREY COPAS, SUNNI STANGL, GINA MINEAR, VELMA MCMILLIAN"
CECILY,,REESE,1,[test],2,F,Chinese,"TREY COPAS, ARIEL PUCCI, TOWANDA BANGS, GEORGIANA VANCLEAVE, SHANTA OBERMILLER"
CHONG,,FLOURNOY,1,[test],2,M,French,"EVAN RAPPAPORT, HILARIO MCLARTY, CLIFFORD ELLINGSWORTH, TYREE BRECHT, TERRESA NAU"
DANIEL,,FRISK,1,[test],2,M,Chinese,"SANTOS YIP, MARGARITO CARBONE, QIANA BOURLAND, ALYSIA TURNBULL, PEGGIE COSNER"
DARLINE,,NICHOLAS,1,[test],3,F,Chinese,"EVAN RAPPAPORT, ROSARIO MANOS, FRANK PROUDFOOT, ELYSE PINNER, MAGEN EVENSEN"
DONELLA,,LAZARD,1,[test],3,F,French,"DARRON POULSEN, WARD LEDERER, ERNESTO SICA, DARREN LEVIN, DEENA BERTI"
EBONI,,GIBBENS,1,[test],2,F,French,"DALTON GREENLEE, NATHANAEL ENTREKIN, DAVID PITTEr, ELYSE PINNER, MARCELINE STROHL"
EDYTH,,AGUAYO,1,[test],3,F,Chinese,"LEO MEFFORD, ERNESTO SICA, ARIEL PUCCI, EUNA FORWARD, MEGAN BOWLEY"
ELLIOT,,HELFRICH,1,[test],1,M,French,"DARREN LEVIN, VELVA LITCHFIELD, EDYTH AGUAYO, ARTIE FOBBS, MEGAN BOWLEY"
ELYSE,,PINNER,1,[test],1,F,Chinese,"ERNESTO SICA, JOAQUIN ELKIN, EBONI GIBBENS, DANILLE FRANZONI, DARLINE NICHOLAS"
JOAQUIN,,ELKIN,1,[test],3,M,French,"LEONARD SISCO, ELYSE PINNER, CAROLE CAUDILL, FRANCINE MERLOS, DEENA BERTI"
LAI,,FINE,1,[test],2,F,Chinese,"CLIFFORD ELLINGSWORTH, DORATHY HURWITZ, ARTIE FOBBS, DONELLA LAZARD, SHANTA OBERMILLER"
LAWERENCE,,ARGUELLES,1,[test],3,M,French,"ARNOLDO MELITO, CLIFFORD ELLINGSWORTH, ARDELLA ALBERTSON, ARTIE FOBBS, DEENA BERTI"
LEO,,MEFFORD,1,[test],3,M,French,"ELTON COKLEY, FRANK PROUDFOOT, EDYTH AGUAYO, TOWANDA BANGS, CHEYENNE RAVEN"
LEONARD,,SISCO,1,[test],1,M,Chinese,"DEL SECKMAN, DARREN LEVIN, DAVID PITTEr, TANIKA SEAGRAVES, GINA MINEAR"
MAGEN,,EVENSEN,1,[test],2,F,Chinese,"DALTON GREENLEE, TROY LAICHE, STAN CLAUDIO, BRAIN BAZEMORE, DARLINE NICHOLAS"
MEGAN,,BOWLEY,1,[test],1,F,French,"ELLIOT HELFRICH, LEO MEFFORD, JOHNATHAN EDELSON, KERA ARSENAULT, EDYTH AGUAYO"
QIANA,,BOURLAND,1,[test],1,F,French,"DEL SECKMAN, DANIEL FRISK, ADRIANA MCKITTRICK, ARTIE FOBBS, SHANTA OBERMILLER"
ROMONA,,RUFUS,1,[test],3,F,French,"MORRIS DORAIS, LEO MEFFORD, ARIEL PUCCI, CARROL BEALER, KERA ARSENAULT"
SHANTA,,OBERMILLER,1,[test],2,F,French,"JARRETT DEPAOLI, EBONI GIBBENS, QIANA BOURLAND, LAI FINE, PEGGIE COSNER"
TYREE,,BRECHT,1,[test],3,M,French,"FRANK PROUDFOOT, DEL SECKMAN, SUNNI STANGL, ALISE COLMENERO, DEENA BERTI"
WAI,,TILLISON,1,[test],1,F,Chinese,"EVAN RAPPAPORT, LESLIE SPEER, ELLIS MOSTELLER, DANIEL FRISK, MARYJANE SIPPEL"
XAVIER,,CHEESMAN,1,[test],3,M,French,"LEONARD SISCO, FRANCINE MERLOS, EUNA FORWARD, DARLINE NICHOLAS, CECILY REESE"
ADRIANA,,MCKITTRICK,2,[test],3,F,French,"JOHNATHAN EDELSON, MARCELINE STROHL, FRANCINE MERLOS, DONELLA LAZARD, ISA ETHERTON"
BELIA,,CORTESE,2,[test],2,F,French,"HILARIO MCLARTY, ARIEL PUCCI, DORATHY HURWITZ, VELVA LITCHFIELD, CECILY REESE"
CAROLE,,CAUDILL,2,[test],2,F,French,"ERNESTO SICA, JOAQUIN ELKIN, WERNER PERREIRA, ELLIS MOSTELLER, EVELIA KINSELLA"
CARROL,,BEALER,2,[test],1,M,Chinese,"RUBEN GAO, DAVID PITTEr, DORATHY HURWITZ, CAROLE CAUDILL, TOWANDA BANGS"
DARREN,,LEVIN,2,[test],2,M,Chinese,"HASSAN BALKE, ELLIS MOSTELLER, BRAIN BAZEMORE, LEONARD SISCO, FLORANCE LAPEYROUSE"
DARRON,,POULSEN,2,[test],2,M,French,"JARRETT DEPAOLI, HASSAN BALKE, TAMEKIA TIMMINS, DONELLA LAZARD, OPHELIA LETOURNEAU"
DAVID,,PITTEr,2,[test],2,M,Chinese,"LEONARD SISCO, EBONI GIBBENS, MAGEN EVENSEN, BELIA CORTESE, MELISSIA MCGLOTHLEN"
```



EVELIA,,KINSELLA,2,[test],2,F,Chinese,"ARNOLDO MELITO, HILARIO MCLARTY, DEXTER WELLING, PAT SOWELL, CAROLE CAUDILL"  
FLORANCE,,LAPEYROUSE,2,[test],1,F,French,"TROY LAICHE, DARREN LEVIN, PEARLENE HOCKER, TANIKA SEAGRAVES, CECILY REESE"  
GEORGIANA,,VANCLEAVE,2,[test],1,F,Chinese,"SANTOS YIP, FRANCINE MERLOS, EDYTH AGUAYO, TANIKA SEAGRAVES, CECILY REESE"  
GINA,,MINEAR,2,[test],2,F,French,"JOHNATHAN EDELSON, STAN CLAUDIO, DAVID PITTE, VELVA LITCHFIELD, ROMONA RUFUS"  
HASSAN,,BALKE,2,[test],2,M,French,"SANTOS YIP, LIONEL HEYEN, DARREN LEVIN, EDYTH AGUAYO, ENA DANGELO"  
HILARIO,,MCLARTY,2,[test],2,M,French,"DEL SECKMAN, BERNIE REGINA, DORATHY HURWITZ, ARDELLA ALBERTSON, EVELIA KINSELLA"  
KERA,,ARSENALTY,2,[test],3,F,French,"WARD LEDERER, DEL SECKMAN, JOHNATHAN EDELSON, ROMONA RUFUS, MEGAN BOWLEY"  
LESLIE,,SPEER,2,[test],2,M,Chinese,"MORRIS DORAIS, LIONEL HEYEN, BRAIN BAZEMORE, DAVID PITTE, WAI TILLISON"  
LIONEL,,HEYEN,2,[test],2,M,Chinese,"ELTON COKLEY, ARNOLDO MELITO, ROSARIO MANOS, ARIEL PUCCI, CHONG FLOURNOY"  
MARYJANE,,SIPPEL,2,[test],2,F,French,"MORRIS DORAIS, DEXTER WELLING, TAMEKIA TIMMINS, ROMONA RUFUS, EDYTH AGUAYO"  
OPHELIA,,LETOURNEAU,2,[test],2,F,French,"DARRON POULSEN, ELLIOT HELFRICH, DYLAN BIEN, JOAQUIN ELKIN, LIONEL HEYEN"  
PAT,,SOWELL,2,[test],3,M,French,"EVAN RAPPAPORT, MURRAY KULAS, STAN CLAUDIO, EBONI GIBBENS, EVELIA KINSELLA"  
PEARLENE,,HOCKER,2,[test],2,F,Chinese,"JOAQUIN ELKIN, TROY LAICHE, PAT SOWELL, KENYETTA BASSIN, GINA MINEAR"  
ROSARIO,,MANOS,2,[test],2,M,Chinese,"MURRAY KULAS, ARIEL PUCCI, MARGARITO CARBONE, LIONEL HEYEN, SUNNI STANGL"  
STAN,,CLAUDIO,2,[test],1,M,Chinese,"ELTON COKLEY, PAT SOWELL, ARDELLA ALBERTSON, GINA MINEAR, ARTIE FOBBS"  
TAMEKIA,,TIMMINS,2,[test],2,F,Chinese,"DARRON POULSEN, ARNOLDO MELITO, JARRETT DEPAOLI, MARYJANE SIPPEL, DONELLA LAZARD"  
TOMEKA,,EADIE,2,[test],1,F,French,"DEXTER WELLING, CAROLE CAUDILL, MELISSIA MCGLOTHLEN, ARTIE FOBBS, ISA ETHERTON"  
WARD,,LEDERER,2,[test],2,M,French,"HILARIO MCLARTY, ARIEL PUCCI, JOHNATHAN EDELSON, DORATHY HURWITZ, DONELLA LAZARD"  
ALYSIA,,TURNBULL,3,[test],2,F,French,"ERNESTO SICA, DANIEL FRISK, LIONEL HEYEN, VELVA LITCHFIELD, MELISSIA MCGLOTHLEN"  
ARTIE,,FOBBS,3,[test],3,F,Chinese,"PEARLENE HOCKER, ARDELLA ALBERTSON, LAI FINE, SHANIKA LAMOREAUX, TOMEKA EADIE"  
BERNIE,,REGINA,3,[test],3,M,Chinese,"ARIEL PUCCI, ELLIS MOSTELLER, ARDELLA ALBERTSON, AYSIA TURNBULL, SHANTA OBERMILLER"  
CHEYENNE,,RAVER,3,[test],2,F,French,"LEO MEFFORD, NATHANAELENTREKIN, DORATHY HURWITZ, MARCELINE STROHL, ARDELLA ALBERTSON"  
CLIFFORD,,ELLINGSWORTH,3,[test],2,M,Chinese,"FRANK PROUDFOOT, CAROL BEALER, CHONG FLOURNOY, CAROLE CAUDILL, ENA DANGELO"  
DALTON,,GREENLEE,3,[test],2,M,Chinese,"MURRAY KULAS, JARRETT DEPAOLI, XAVIER CHEESMAN, EBONI GIBBENS, MAGEN EVENSEN"  
DANILLE,,FRANZONI,3,[test],2,F,Chinese,"MORRIS DORAIS, FRANK PROUDFOOT, ELLIS MOSTELLER, ZANE GAMBREL, ELYSE PINNER"  
DEL,,SECKMAN,3,[test],2,M,French,"HILARIO MCLARTY, TYREE BRECHT, LEONARD SISCO, QIANA BOURLAND, OPHELIA LETOURNEAU"  
ELLIS,,MOSTELLER,3,[test],2,M,French,"DARREN LEVIN, LEONARD SISCO, WAI TILLISON, CAROLE CAUDILL, DANILLE FRANZONI"  
ELTON,,COKLEY,3,[test],1,M,Chinese,"SANTOS YIP, LIONEL HEYEN, DORATHY HURWITZ, FRANCINE MERLOS, TOMEKA EADIE"  
ENA,,DANGELO,3,[test],2,F,Chinese,"CLIFFORD ELLINGSWORTH, MARGARITO CARBONE, ELLIS MOSTELLER, BELIA CORTESE, FRANCINE MERLOS"  
ERNESTO,,SICA,3,[test],2,M,French,"ELLIOT HELFRICH, ELYSE PINNER, CAROLE CAUDILL, EDYTH AGUAYO, AYSIA TURNBULL"  
ISA,,ETHERTON,3,[test],1,F,French,"DARRON POULSEN, CAROL BEALER, ADRIANA MCKITTRICK, ALISE COLMENERO, TOMEKA EADIE"  
JARRETT,,DEPAOLI,3,[test],2,M,Chinese,"WERNER PERREIRA, AYSIA TURNBULL, TANIKA SEAGRAVES, SHANTA OBERMILLER, PEGGIE COSNER"  
JOHNATHAN,,EDELSON,3,[test],2,M,French,"EVAN RAPPAPORT, JARRETT DEPAOLI, STAN CLAUDIO, GINA MINEAR, MEGAN BOWLEY"  
KENYETTA,,BASSIN,3,[test],3,F,Chinese,"GUILLERMO GAULIN, JARRETT DEPAOLI, CLIFFORD ELLINGSWORTH, EUNA FORWARD, DARLINE NICHOLSON"  
MARCELINE,,STROHL,3,[test],2,F,French,"ARNOLDO MELITO, DEXTER WELLING, ADRIANA MCKITTRICK, CAROLE CAUDILL, CHEYENNE RAVEN"  
PEGGIE,,COSNER,3,[test],2,F,French,"DALTON GREENLEE, JARRETT DEPAOLI, DANIEL FRISK, FRANCINE MERLOS, KENYETTA BASSIN"  
RUBEN,,GAO,3,[test],2,M,French,"MURRAY KULAS, CAROL BEALER, EDYTH AGUAYO, MELISSIA MCGLOTHLEN, ALISE COLMENERO"  
SANTOS,,YIP,3,[test],2,M,Chinese,"ELTON COKLEY, DEL SECKMAN, HASSAN BALKE, DANIEL FRISK, GEORGIANA VANCLEAVE"  
SHANIKA,,LAMOREAUX,3,[test],3,F,Chinese,"DEXTER WELLING, JASON RODD, MARYJANE SIPPEL, GINA MINEAR, ALISE COLMENERO"  
TANIKA,,SEAGRAVES,3,[test],3,F,Chinese,"ROSARIO MANOS, DYLAN BIEN, JARRETT DEPAOLI, LEONARD SISCO, GEORGIANA VANCLEAVE"  
TROY,,LAICHE,3,[test],1,M,French,"ELLIOT HELFRICH, STAN CLAUDIO, MAGEN EVENSEN, CAROLE CAUDILL, PEARLENE HOCKER"  
VELMA,,MCMILLIAN,3,[test],3,F,Chinese,"MORRIS DORAIS, ARNOLDO MELITO, BRAIN BAZEMORE, ELYSE PINNER, DORATHY HURWITZ"  
VELVA,,LITCHFIELD,3,[test],3,F,Chinese,"ARNOLDO MELITO, TYREE BRECHT, KENYETTA BASSIN, AYSIA TURNBULL, GINA MINEAR"  
ZANE,,GAMBREL,3,[test],3,M,French,"LEO MEFFORD, GUILLERMO GAULIN, LIONEL HEYEN, FRANCINE MERLOS, DANILLE FRANZONI"  
ARIEL,,PUCCI,4,[test],2,M,Chinese,"WARD LEDERER, HILARIO MCLARTY, FRANK PROUDFOOT, LIONEL HEYEN, ROMONA RUFUS"  
ARNOLDO,,MELITO,4,[test],1,M,French,"LIONEL HEYEN, MARCELINE STROHL, VELVA LITCHFIELD, MEGAN BOWLEY, VELMA MCMILLIAN"  
DEENA,,BERTI,4,[test],3,F,French,"ARNOLDO MELITO, LAWRENCE ARGUELLES, BELIA CORTESE, CAROLE CAUDILL, SHANTA OBERMILLER"  
DEXTER,,WELLING,4,[test],2,M,French,"DEL SECKMAN, MARCELINE STROHL, SHANIKA LAMOREAUX, TOMEKA EADIE, STEPHENIE MADILL"



DORATHY,,HURWITZ,4,[test],1,F,Chinese,"HILARIO MCLARTY, TREY COPAS, CARROL BEALER, BELIA CORTESE, CHEYENNE RAVER"  
DYLAN,,BIEN,4,[test],2,M,Chinese,"DORATHY HURWITZ, ARDELLA ALBERTSON, FLORANCE LAPEYROUSE, OPHELIA LETOURNEAU, ISA ETHERTON"  
EUNA,,FORWARD,4,[test],1,F,French,"XAVIER CHEESMAN, ARDELLA ALBERTSON, KENYETTA BASSIN, TOWANDA BANGS, TOMIKA EADIE"  
EVAN,,RAPPAPORT,4,[test],1,M,French,"ELLIOT HELFRICH, CHONG FLOURNOY, ADRIANA MCKITTRICK, WAI TILLISON, DARLINE NICHOLAS"  
FRANCINE,,MERLOS,4,[test],1,F,French,"ELTON COKLEY, ROSARIO MANOS, JOAQUIN ELKIN, ZANE GAMBREL, ENA DANIELLO"  
FRANK,,PROUDFOOT,4,[test],1,M,Chinese,"ARIEL PUCCI, TYREE BRECHT, BERNIE REGINA, ELYSE PINNER, DANILLE FRANZONI"  
GUILLERMO,,GAULIN,4,[test],1,M,French,"HILARIO MCLARTY, ARIEL PUCCI, ZANE GAMBREL, ARTIE FOBBS, STEPHENIE MADILL"  
JASON,,RODD,4,[test],2,M,Chinese,"GUILLERMO GAULIN, TYREE BRECHT, ZANE GAMBREL, LAI FINE, ISA ETHERTON"  
KIETH,,POEHLER,4,[test],1,M,Chinese,"DARRON POULSEN, ELLIOT HELFRICH, DEL SECKMAN, DALTON GREENLEE, KENYETTA BASSIN"  
MARGARITO,,CARBONE,4,[test],1,M,Chinese,"ROSARIO MANOS, DANIEL FRISK, DARREN LEVIN, SHANTA OBERMILLER, ENA DANIELLO"  
MELISSA,,MCGLOTHLEN,4,[test],2,F,French,"RUBEN GAO, BERNIE REGINA, DAVID PITTER, DORATHY HURWITZ, ALISE COLMENERO"  
MORRIS,,DORAIS,4,[test],1,M,French,"JOAQUIN ELKIN, LESLIE SPEER, LAI FINE, MEGAN BOWLEY, VELMA MCMILLIAN"  
MURRAY,,KULAS,4,[test],1,M,Chinese,"ROSARIO MANOS, DALTON GREENLEE, RUBEN GAO, GEORGIANA VANCELEAVE, ENA DANIELLO"  
NATHANAEL,,ENTREKIN,4,[test],2,M,Chinese,"EVAN RAPPAPORT, WERNER PERREIRA, EBONI GIBBENS, CAROLE CAUDILL, ARDELLA ALBERTSON"  
STEPHENIE,,MADILL,4,[test],3,F,French,"GUILLERMO GAULIN, JOHNATHAN EDELSON, DEXTER WELLING, ZANE GAMBREL, ADRIANA MCKITTRICK"  
SUNNI,,STANGL,4,[test],2,F,Chinese,"ROSARIO MANOS, TYREE BRECHT, BRAIN BAZEMORE, DANILLE FRANZONI, ISA ETHERTON"  
TERRESA,,NAU,4,[test],2,F,French,"SUNNI STANGL, DANILLE FRANZONI, MEGAN BOWLEY, DARLINE NICHOLAS, CECILY REESE"  
TOWANDA,,BANGS,4,[test],3,F,Chinese,"DEL SECKMAN, LEO MEFFORD, CARROL BEALER, EUNA FORWARD, CECILY REESE"  
TREY,,COPAS,4,[test],1,M,French,"BRAIN BAZEMORE, ELYSE PINNER, DORATHY HURWITZ, TANIKA SEAGRAVES, CHEYENNE RAVER"  
WERNER,,PERREIRA,4,[test],3,M,French,"ROSARIO MANOS, ZANE GAMBREL, MAGEN EVENSEN, MARCELINE STROHL, CAROLE CAUDILL"

Second file is named "Year 3 - Creating classes for the new school year - exclusions and inclusions.csv". The content of the file is

Exclusion or Inclusion,Students  
Must NOT be in the same class,"KIETH POEHLER, NATHANAEL ENTREKIN"  
Must NOT be in the same class,"HASSAN BALKE, WARD LEDERER"  
Must NOT be in the same class,"DANILLE FRANZONI, VELVA LITCHFIELD"  
Must NOT be in the same class,"BELIA CORTESE, GUILLERMO GAULIN"