# Django School Information System

*Release*

December 09, 2014

# Contents

**Date** November 21, 2013

Django-SIS is an open source school information system built with Django. It relies heavily on the django admin interface for backend usage (registrar, etc.) and is intended for schools with or without work study programs. Django-SIS is able to integrate with Naviance Premium Accounts for college preparedness, Engrade for grades, SugarCRM for sales and customer tracking, and National Student Clearinghouse for tracking alumni.

In sum, Django-SIS includes pluggable apps designed to cover most if not all a school's needs. The apps include: School Information, Admissions, Alumni, Attendance, Discipline, Schedules/Courses/Grades, Volunteer Tracking, and Work-Study.

The purpose of this documentation is to be a user manual for end users. Most features will be highlighted and discussed, along with explanations and how-tos for everyday tasks. If you are a developer more interested in the technical aspects of Django-SIS, please refer to our github page.

CHAPTER 1

# Administrator Information

**Libreoffice on the server**

Should be run as an upstart job like /etc/init/libreoffice.conf:

```
start on runlevel 2

exec /usr/lib/libreoffice/program/soffice.bin '-accept=socket,host=localhost,port=2002;urp;StarOffice

respawn
respawn limit 10 120
```

Then start with "start libreoffice"

## 1.1 User Permissions

Django-SIS allows administrators to control individual user permissions. To simplify this process, Django-SIS groups individual permissions into larger groups which the administrator can designate accordingly. Some of the groups are as follows:

**Teachers**: Users with this designation may create tests, view students, enter grades, and take attendance.

**Counseling**: record student meetings, refer students, and list follow-up actions

**Faculty**: view alumni, students, and mentoring information

**Work Study**: view work study attendance, fees, visits, companies, payment options, contact supervisors, time sheets, surveys, assign work teams and work team users.

**Registrar**: edit templates, view applicants, edit admissions, view/edit attendance, add custom fields, sync Endgrade courses, create schedules, reports, transcript notes, and school years.

**Volunteer**: add/change/delete volunteer hours, sites, supervisors, and student volunteers

It is possible to assign individual user permissions that are found in one group to an individual user that only has permissions from another group. For example, you can assign a teacher (who only has teacher permissions) the ability to view a student's counseling records or work study information. This allows school administrators to create unique users with flexible permissions. Further, administrators can create superusers who have permissions from all groups.

## 1.2 Importing Data Into Sword

Django-SIS supports the import of data into its database.

In order to make the import process as simple as possible for schools transitioning into Django-SIS or schools preparing for the new school year, Django-SIS allows data to be imported via Excel or LibreOffice documents.

Keep in mind that this section is for **importing** data, and not *updating* data. This section assumes that students, applicants, etc. do not yet exist in Django-SIS (i.e.: new incoming freshmen or new applicants).

If you want to use simple import to mass update information (i.e: assigning existing student workers work days and placements). Then please refer to the next chapter.

**Before you Import**

Prior to importing data, you will need an Excel spreadsheet with information pertinent to the model you are attempting to import. Set column headers accordingly. Django-SIS is able to import all information that is able to be entered manually in a field.

Keep in mind, the list below would be on a per import basis, and would be on a single Excel workbook. You cannot for example, import students and applicants in different workbooks. Save a new Excel file for each; the fields below will be the column headers on your Excel workbook, and the root models for import would be what's in italics. Here's a reference list of some column headers you can use for importing the most common models:

*Students*

Unique ID, First Name ,Last Name, username, grad date, Student cell ph, middle name, class of year, STUDENT PHONE, GENDER, BIRTH DATE, Social Security, Student E-Mail, Alert, Primary Cohort, Parent email, homeroom, preferred language, picture, password

NOTE: First name, Last Name, and username/unique ID are required for importing new students.

*Student Workers*

First Name, Last Name, middle name, Username, notes, ADP Number, alert, alt email, am route, applicant, birth date, city, class of year, cohort, course, date dismissed, working day, placement, email, parent email, student pay rate, sex, SSN

*Applicants*

First name, Last name, middle name, birth date, present school, heard about us, first contact, withdrawn note, total income, adjusted available income, application decsion, application decision by, SSN, Sex, Ethnicity, Religion, Place of Worship, Year, School year, HS grad yr, Elem grad yr, email, notes, country of birth, family preferred language, lives with

*Marking Period*

Name, shorname, Start Date, End date, grades due, School year, weight (integer. ex. 1), active (=TRUE, or =FALSE), show reports (=TRUE, or =FALSE).

NOTE: name, shortname, start-end date, school year and weight are required fields.

*Courses*

Fullname, shortname, Marking Period, description, teacher (their username), homeroom (=TRUE, or =FALSE), Graded (=TRUE, or =FALSE), credits (integer field, ex. 1.0), department, level (Freshman, Sophomore, Junior, Senior).

NOTE: only fullname and shortname are required.

*Location*

Name

NOTE: This generally refers to a classroom name or location. Ex. Room 343

*Period*

Name, Start time, End time

*Family preferred Language*

Name, iso code (optional)

*Class Year*

Year (number- ex. 2014), Full name (ex. Class of 2014)

*Faculty*

Importing faculty essentially creates a user and assigns them the “faculty” permission. Designating =TRUE to the ‘teacher’ field just adds the ‘teacher’ permissions.

Username, First name, Last Name, email address, number, ext, Teacher (=TRUE or =FALSE)

NOTE: only the username field is required.

*Company*

This is for schools utilizing the CWSP module.

Name

*Work Team*

For schools setting up CWSP.

Team name, job description, company description, login (username), paying (=TRUE or =FALSE), Address, City, State, Zip, Travel route, Directions to, Directions pickup

The only required field is team name.

*Volunteer Site Location*

Only for schools planning to use the Volunteer module.

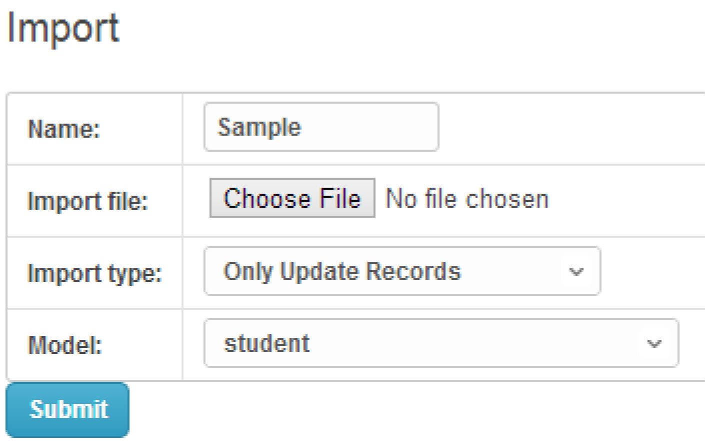
Site name, Site address, Site city, Site state, Site zip

NOTE: Django-SIS will guess matches based off your column headers, so if you enter First Name or fname, or FiRSt NaME as a header on your Excel doc, Django-SIS will determine which field that refers to.

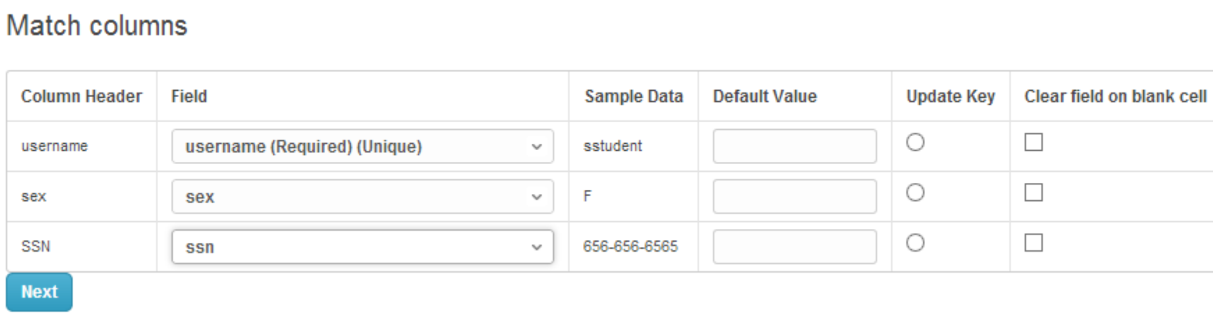
**How To Import Data**

Importing data *requires* the appropiate permissions for the user. The method is described below:

1. Select Admin > School import from the navigation menu.
2. Enter a name for the import (can be anything).
3. Select browse to locate your Excel document
4. Under Import type, select from Create New Records, Create and Update Records, and Only Update Records.
5. Select a Model. This refers to where you are importing the data. Select students for students, applicants for applicants, etc.




6. Click Submit.
7. The next page will give users a preview of what their import will look like. Django-SIS tries to match the column headers in your document with an available field in Django-SIS. You can always edit the field (via drop down box) if Django-SIS displays an incorrect field.




8. After you've matched all the fields to the sample data (i.e. SSN field actually displays a SSN in the sample data column), you can simulate the import or run it. Simulating the import won't actually import the data, but it will let you know in advance if there are any issues with what you're trying to import.

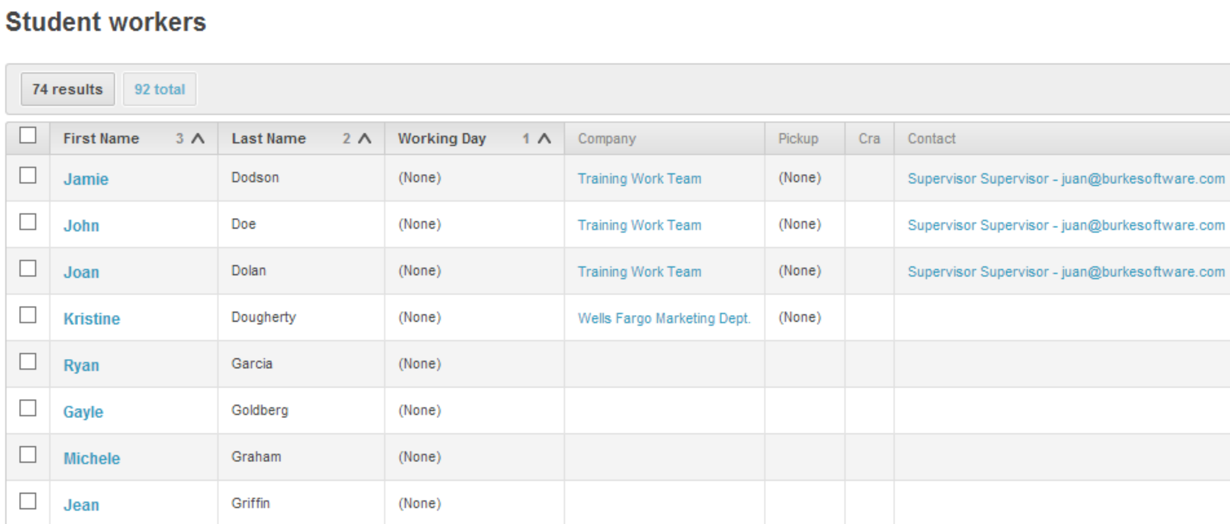
## 1.3 Updating data using Simple Import

Some things to keep in mind prior to using the simple import for updating student data.

1. You will **always** need to have an update key. This lets Django-SIS reference what object is being updated. Typically, you will want to use a username, Django-SIS ID or unique ID. Getting IDs for objects is easy. Utilize the export to xls feature from your chosen page (student, student worker, applicant, etc.), and select ID, username or unique ID.
2. Data must already exist for that model. For example, you can't update information on a student that doesn't already exist. You **can** however utilize the "Create and Update" import type to do something like this, but this can't be done on the "Update Only" type.
3. Make sure you select the correct model type.

**Example**

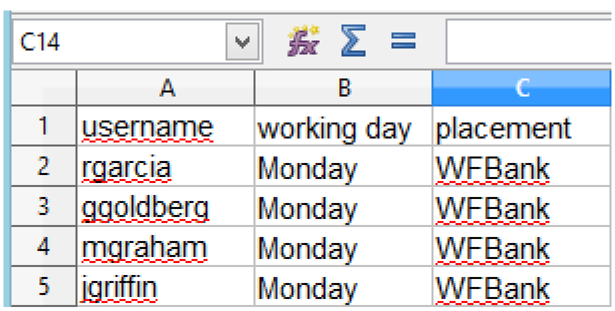
In this example, we will assume that we have a list of new freshmen who were just made into student workers. These student workers will now need their job placements and working days assigned. The picture below highlights the aforementioned:


Student workers

74 results | 92 total

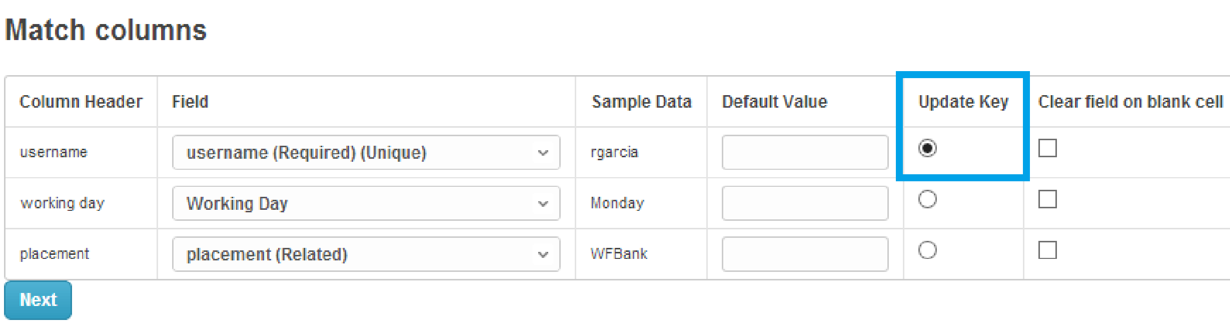
| | First Name 3 | Last Name 2 | Working Day 1 | Company | Pickup | Cra | Contact |
|---|---|---|---|---|---|---|---|
| ☐ | Jamie | Dodson | (None) | Training Work Team | (None) | | Supervisor Supervisor - juan@burkesoftware.com |
| ☐ | John | Doe | (None) | Training Work Team | (None) | | Supervisor Supervisor - juan@burkesoftware.com |
| ☐ | Joan | Dolan | (None) | Training Work Team | (None) | | Supervisor Supervisor - juan@burkesoftware.com |
| ☐ | Kristine | Dougherty | (None) | Wells Fargo Marketing Dept. | (None) | | |
| ☐ | Ryan | Garcia | (None) | | | | |
| ☐ | Gayle | Goldberg | (None) | | | | |
| ☐ | Michele | Graham | (None) | | | | |
| ☐ | Jean | Griffin | (None) | | | | |

1. Get the student worker usernames for update. These usernames will be used as the **update key**. This can be done by checking the box by each student and selecting *export to xls* from the action bar at the bottom of the student worker screen and check off *username*.
2. Using Excel, create a document with the column headers, **username**, **working day**, and **placement**.


C14

| | A | B | C |
|---|---|---|---|
| 1 | username | working day | placement |
| 2 | rgarcia | Monday | WFBank |
| 3 | ggoldberg | Monday | WFBank |
| 4 | mgraham | Monday | WFBank |
| 5 | jgriffin | Monday | WFBank |

3. Access the the simple import page. Admin > School Import
4. At this screen: name your import, select your Excel document, set the import type to **only update records**, and set the model to **student worker**. Click submit.
5. You will be taken to the match columns screen shown below:

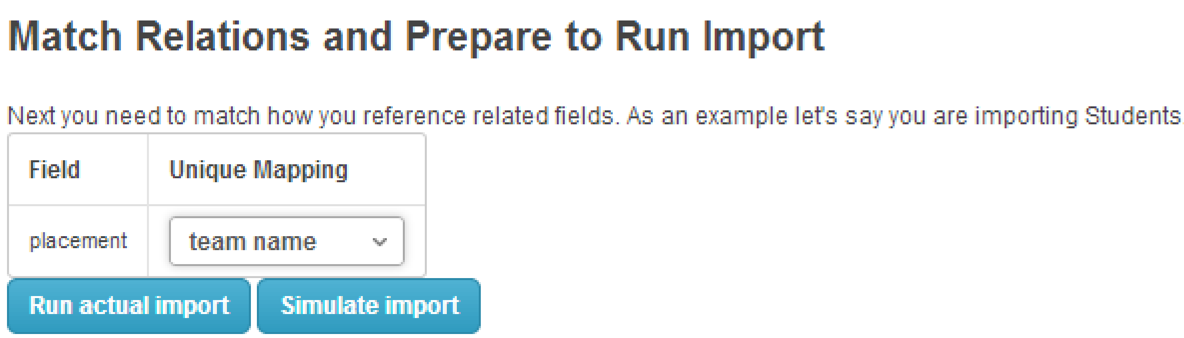

Match columns

| Column Header | Field | Sample Data | Default Value | Update Key | Clear field on blank cell |
|---|---|---|---|---|---|
| username | username (Required) (Unique) | rgarcia | | ◉ | ☐ |
| working day | Working Day | Monday | | ○ | ☐ |
| placement | placement (Related) | WFBank | | ○ | ☐ |

Next

NOTE: Notice in the image above I have selected username as the **update key**. As stated earlier, using either username, ID, or unique ID as the update key is easiest.

6. Click *next*. The match relations screen will ask how to reference a field- this will typically be by a name or ID. In the Excel document for import, we listed a team name for placement so in this example we will use team name instead of ID.




7. Click **Simulate import** to check for errors, then run.
8. Return to the student worker page, and your updated data will be displayed.

| | First Name 3 | Last Name 2 | Working Day 1 | Company |
|---|---|---|---|---|
| | Ryan | Garcia | Monday | WFBank |
| | Gayle | Goldberg | Monday | WFBank |
| | Michele | Graham | Monday | WFBank |
| | Jean | Griffin | Monday | WFBank |

## 1.4 Configurations

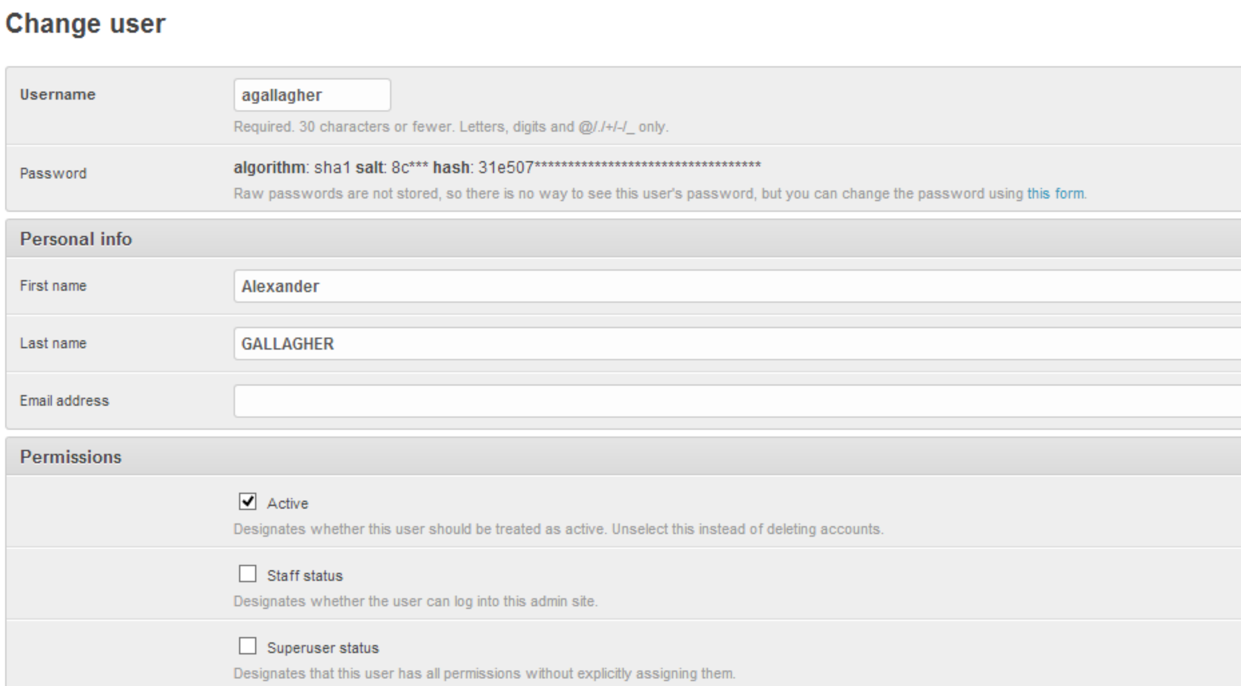
Django-SIS contains a number of built-in configurations that are created with each new instance designed to make functions easier to edit or implement.

For example, in configurations for email in the **How to obtain student email** function, users may designate three values designed to direct Django-SIS emails. **Append** appends the domain name after a student's username like jstudent@domainname.org. **User** takes the email address from the Auth->User record. **Student** takes the email address marked from the *alt email* field of a student record page.

## 1.5 Creating Users

To manually add users, follow the directions below:

1. Under the **Administrators** panel, select Add+ by *Users*
2. Create a username and temporary password.
3. Next, you will need to edit the user. Select the newly created users from the users list.
4. Assign the user a name/last name- and ideally assign them a group, which will be discussed in the next chapter below.




Overall, the groupings are self explanatory- if you're creating a registrar, assign them the **registrar** group, an admissions staff member would be assigned the **admissions** group, etc.

## 1.6 Getting Started

**Server:** Django-SIS can be installed in any platform that can run Django. It should be noted, however, that all testing is done in Ubuntu Linux 10.04 with MySQL.

**Client:** Django-SIS is divided into two parts: the admin site and the student/company-facing site. The student/company-facing site is tested in Firefox, Chrome, Opera, and IE 6,7,8. The admin site is tested only in standards-compliant browsers such as Firefox, Opera, and Chrome. If using IE, you should install the Chrome Frame add-on.

**Editing Templates** requires Office software. Creating report templates require LibreOffice and *must* be saved in ODT format. Keep in mind that end-users may select their preferred office format preference, so ODT is *not* required to just view a report.

Using the ISO-supported Open Document format is recommended for best inter-operability, however doc and xls binary formats are highly supported. In rare cases, formatting may be slightly different in these formats. Office Open XML, while supported, is *not* recommended.

## 1.7 Log Entries

Log entries record all actions completed during a Django-SIS's instance. This allows administrators and superusers to locate any changes made at specific dates or times. Admins will see a dashboard similar to what is shown below:

Log entries

1697 total | 1 | 2 | 3 | 4 ..., 16 | 17

| | User | Action time | Content type | Object repr | Is addition | Is deletion | Is change |
|---|---|---|---|---|---|---|---|
| | admin | April 16, 2013, 11:30 a.m. | report | Report object | False | False | True |
| | admin | April 16, 2013, 11:28 a.m. | report | Report object | False | False | True |
| | juan | April 15, 2013, 4:05 p.m. | admission level | Inquiry | False | False | True |
| | juan | April 15, 2013, 3:58 p.m. | applicant | Crystal Powers | False | False | True |

**User** refers to which user made a change.

**Action time** details the date and time when the change was made.

**Content type** is the model on which the change was made, e.g. applicant, student, etc.

**Object repr** assigns a specific name to the content type. For example, if applicant was the content type, then object repr will list an exact name like Joe Student.

**Is Addition, Is Deletion, Is Change**: True/False indicator which shows what type of action was completed.

Similar to other dashboards in Django-SIS, users may sort by clicking column headers and using the filter tool.

## 1.8 Templates

All Django-SIS instances come packaged with a set of general templates. These templates allow users to generate a number of varied reports, including:Tardy Letters, Daily Attendance, Progress Reports, Transcripts, Travel Maps, Test Results, Discipline Report

A list of all available templates, free to download is found here.

Django-SIS further allows users to create and edit their own templates to be used accordingly and will be discussed in the next section, Report Writing.

## 1.9 Report Writing and Creating Templates

**Note** Before you proceed, please be aware that in most cases it's best to simply edit existing templates found in your templates location, rather than creating entirely new templates as this section will discuss.

Django-SIS provides the means for end users to create and utilize their own customized reports/templates. All reports are made using the Appy Framework.

The basic process works like this: user creates report template in a word processor >>> the template gets uploaded into Django-SIS >>> Download/use finished report.

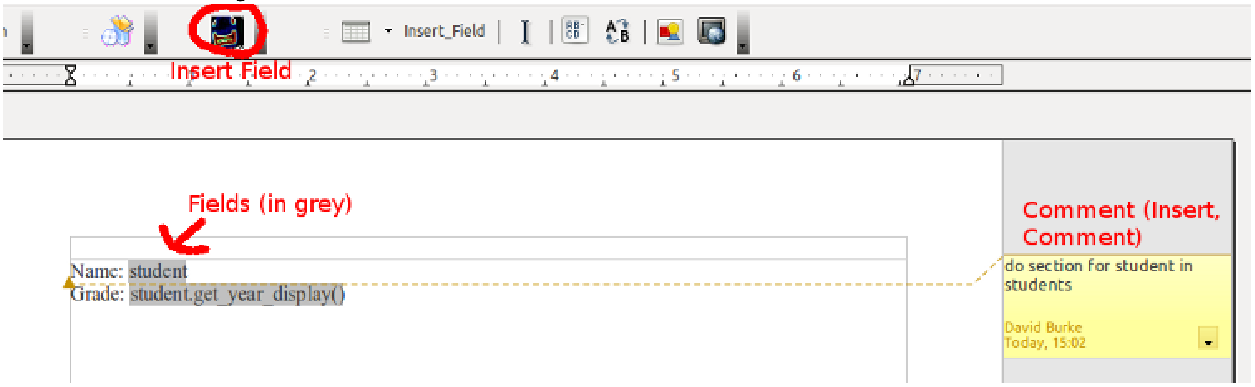
To get started, it is first recommended that you use LibreOffice with the Insert Field extension found here.

**Note:** When creating templates, ODT format is *required* and all files must be saved in the .odt extension, which is the default in LibreOffice.

**Note:** Microsoft Office can be used with track changes used to denote fields, however, this method is *not* recommended.

**Note:** All finished reports may be opened with Microsoft Office.

1. When you enter “fields” in Libre, this refers to database fields.
2. Insert field using the insert field extension mentioned above




3. Edit a field by double clicking on one.

*Note:* You can see a list of available fields to choose from by typing this into your Django-SIS instance’s URL. SAMPLESCHOOLURL/admin/doc/models. Some fields are calculated, for example he_she is based off of the sex of a student. Any type: list field cannot be used directly, but must be placed in a loop.

**Logic in Templates** You may use any Python logic in a template. For example in the above screenshot there is a note “do section for student in students”. This logic can technically be placed in a field, however it’s easier to read in a note. To create a note click Insert > Comment. In the example a section is being created for each student in the field “students”. students is a list of students as defined in “School Reports” in Django-SIS. To create a section click Insert, Section. In the example the section includes a page break. Django-SIS will create a section (page break included) for each student in your list of students. This makes for similar results of a mail merge. You may also “do row” or “do cell” to create tables.

You may even include Django specific code, for example students.filter(fname=”Joe”) would result in a list of students with the first name of “Joe”. For more see‘Django’s retrieving objects <https://docs.djangoproject.com/en/dev/topics/db/queries/#retrieving-objects>‘_. This may get complex fast, therefore Django-SIS offers some basic sorting and filtering options for you. See School Reports with Django-SIS. Essentially School Reports will give you the variable students, with your desired filters. If you selected only one student, you will instead have a “student” variable. From here you usually want some type of logic, such as do section for student in students.

**Spreadsheet Reports** work differently. You can add additional fields to any student related spreadsheet. Select User Preferences and add additional fields here. These additional fields are defined by an administrator and follow the typical . notation (placement.address gets the address of the placement). The gradebook spreadsheet is a special case and a template can be used here. See the included template called “grade spreadsheet”.

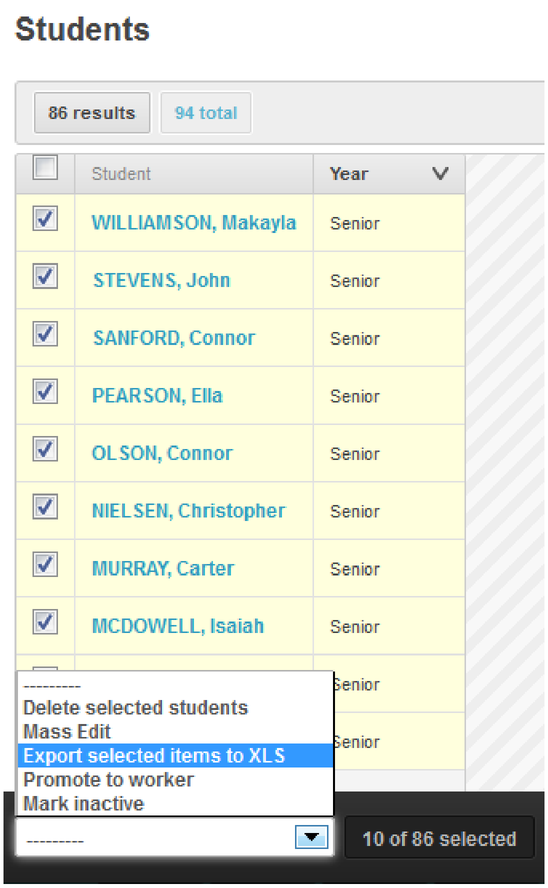
**Database Field Names** Click on Documentation, then Models to view various Database models. You can chain them by placing . to any related fields. For example student.placement.address would yield the address of the placement of that student.

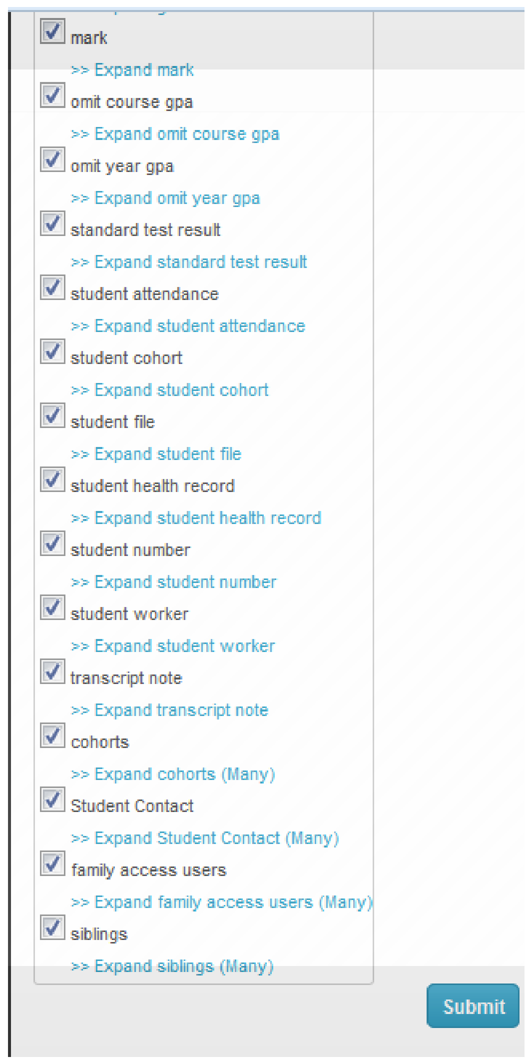
## 1.10 Exporting Django-SIS data to Excel

Django-SIS allows users to export into Excel any and all data that users have input into their respective Django-SIS instance. The process of exporting information is very simple, and detailed below:

1. Click on any model you want to edit from your Django-SIS home dash- ex. students, applicants, student workers, discipline, etc.
2. This will take you to the basic familiar dashboard for that model.

3. Click the checkbox next to each student you want to pull info from.
4. Select the black drop down box located towards the bottom of the page.
5. Select "export to xls"
6. A screen asking what you want to be exported appears- make your selections.
7. Submit.

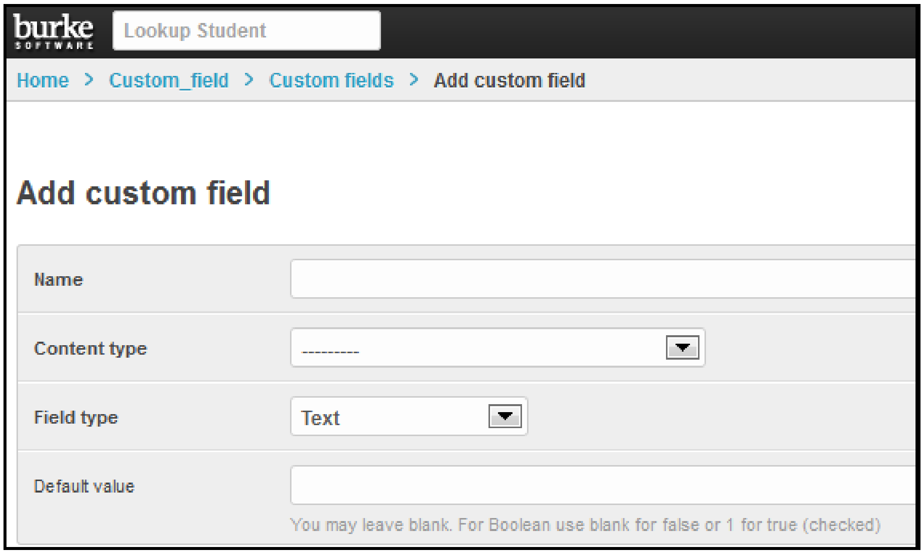



## 1.11 Custom Fields

The custom fields option allows schools additional flexibility with regards to storing additional information to a particular model (student, applicants, student worker, etc.).

Under Admin > Custom Fields, the custom fields creation screen displays:




Required fields:

Name- Refers to the name of the custom field. Note: this name will be visible to other end users

Content Type- Designates which model to affix the custom field to. (Student, Alumni, Applicant, Faculty, etc.)

Field Type- Text, Integer, and Boolean- select the type of custom field.

NB: Boolean refers to a simple checkbox. The box can be checked or unchecked when created based on preference. Leave blank for unchecked and enter "1" for checked under the "Default Value" in the creation screen shown above.

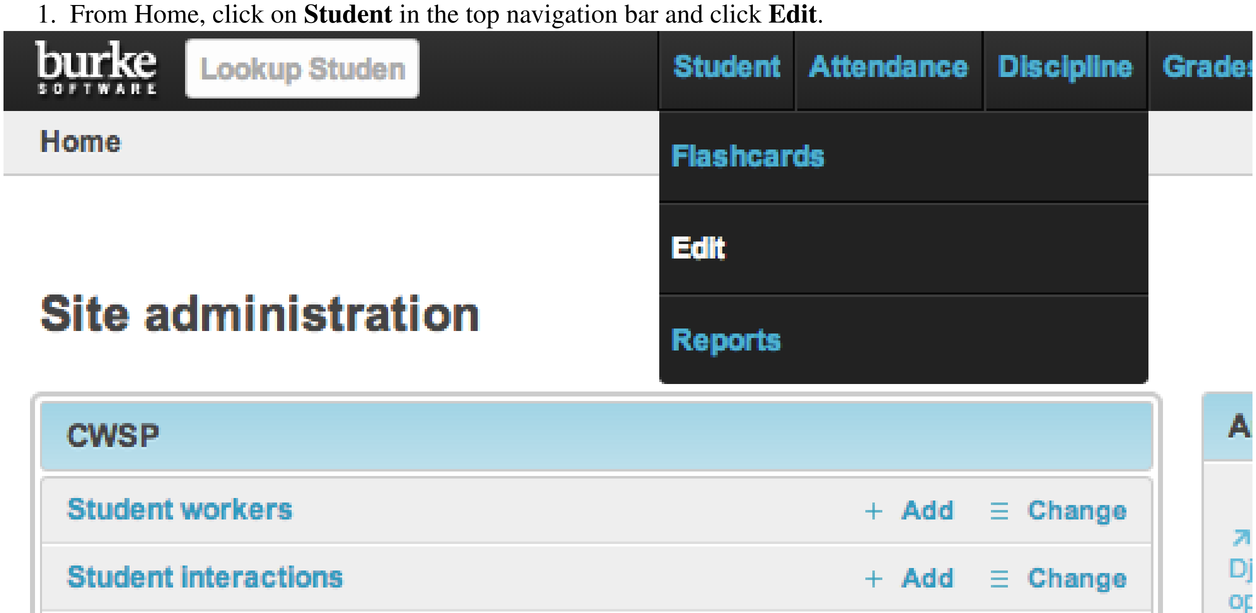
CHAPTER 2

# Student Information

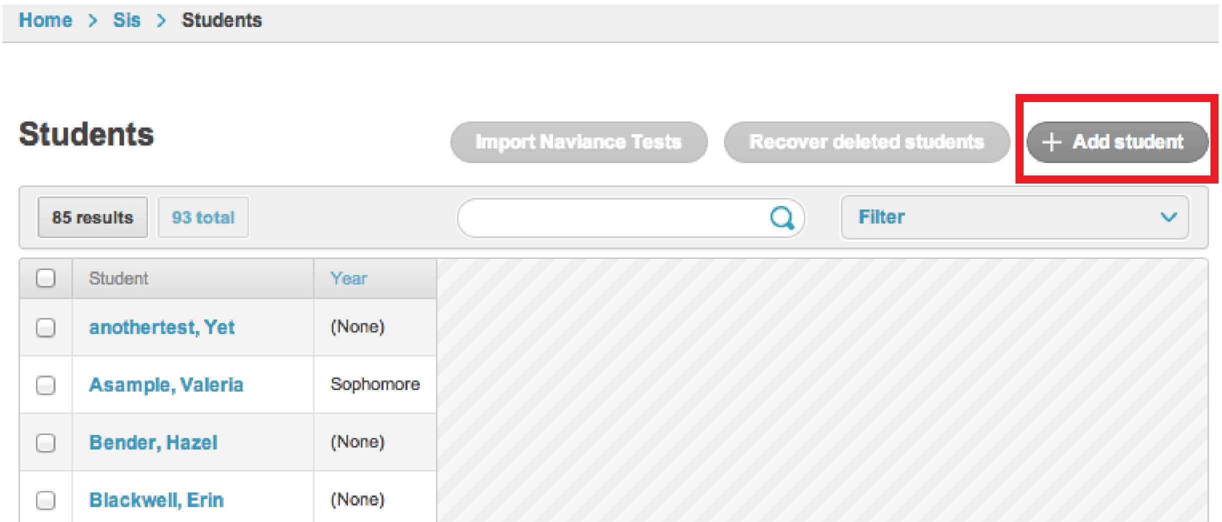
The SIS is the central module of Django-SIS which contains profiles, attendance, discipline, work study, and other details pertaining to the student. For information on admissions, adding students, attendance, and discipline, please follow the pertinent headings.

## 2.1 Adding Students

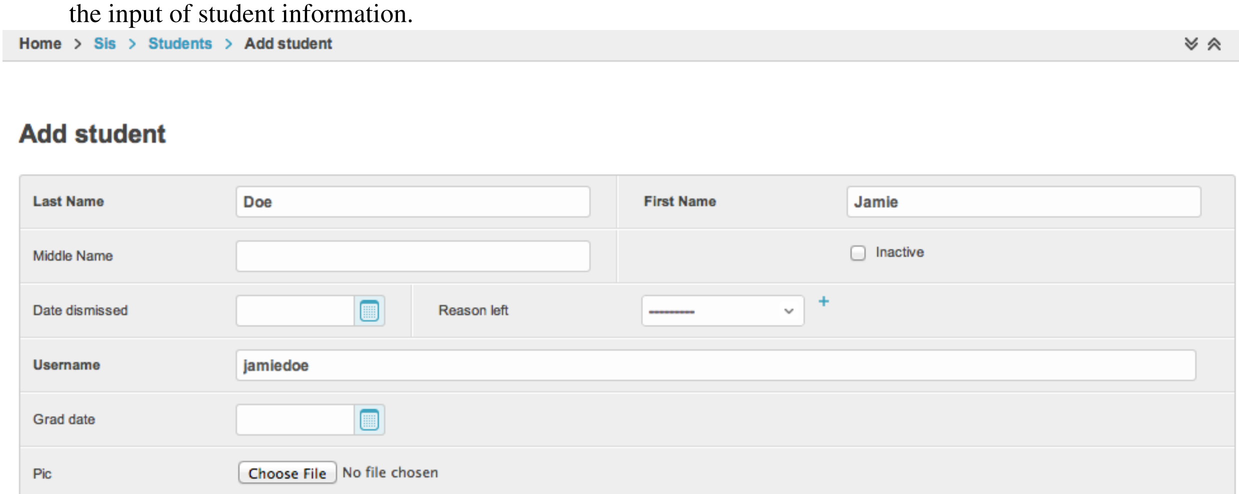
1. From Home, click on **Student** in the top navigation bar and click **Edit**.



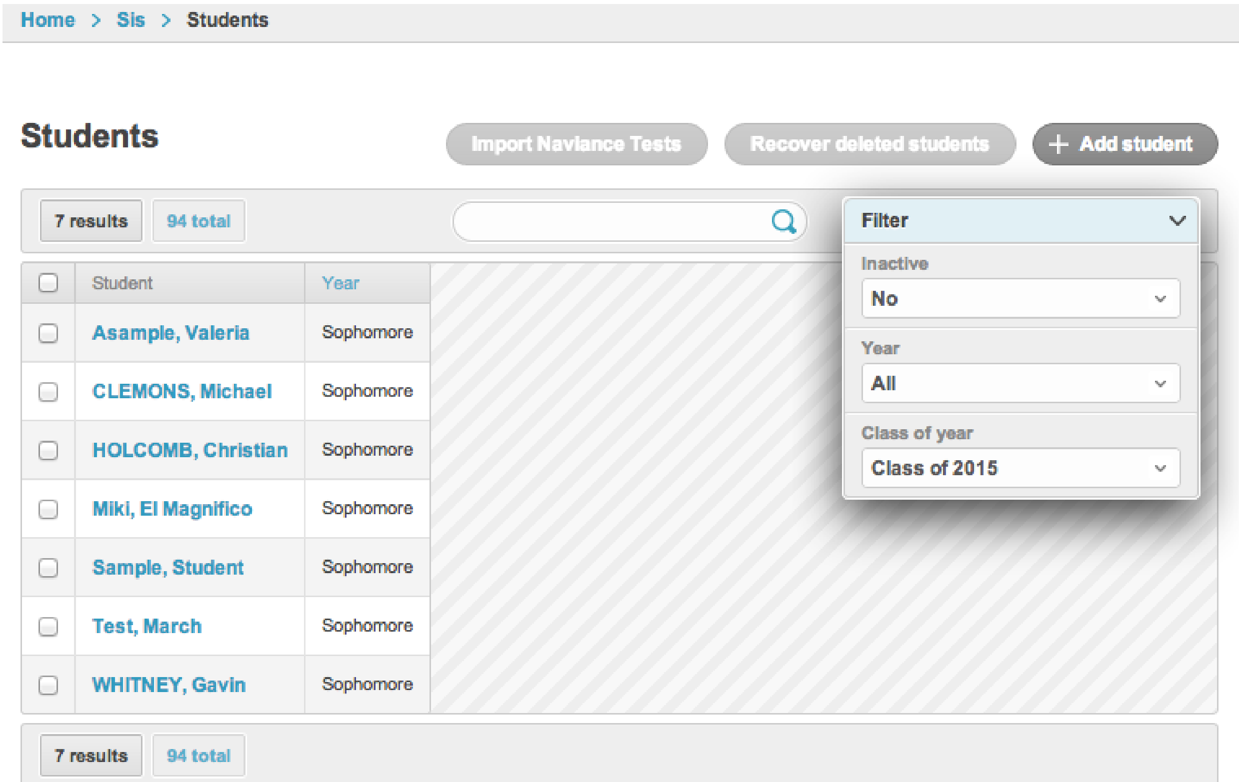

2. On the top right, click the **+ Add student** button.



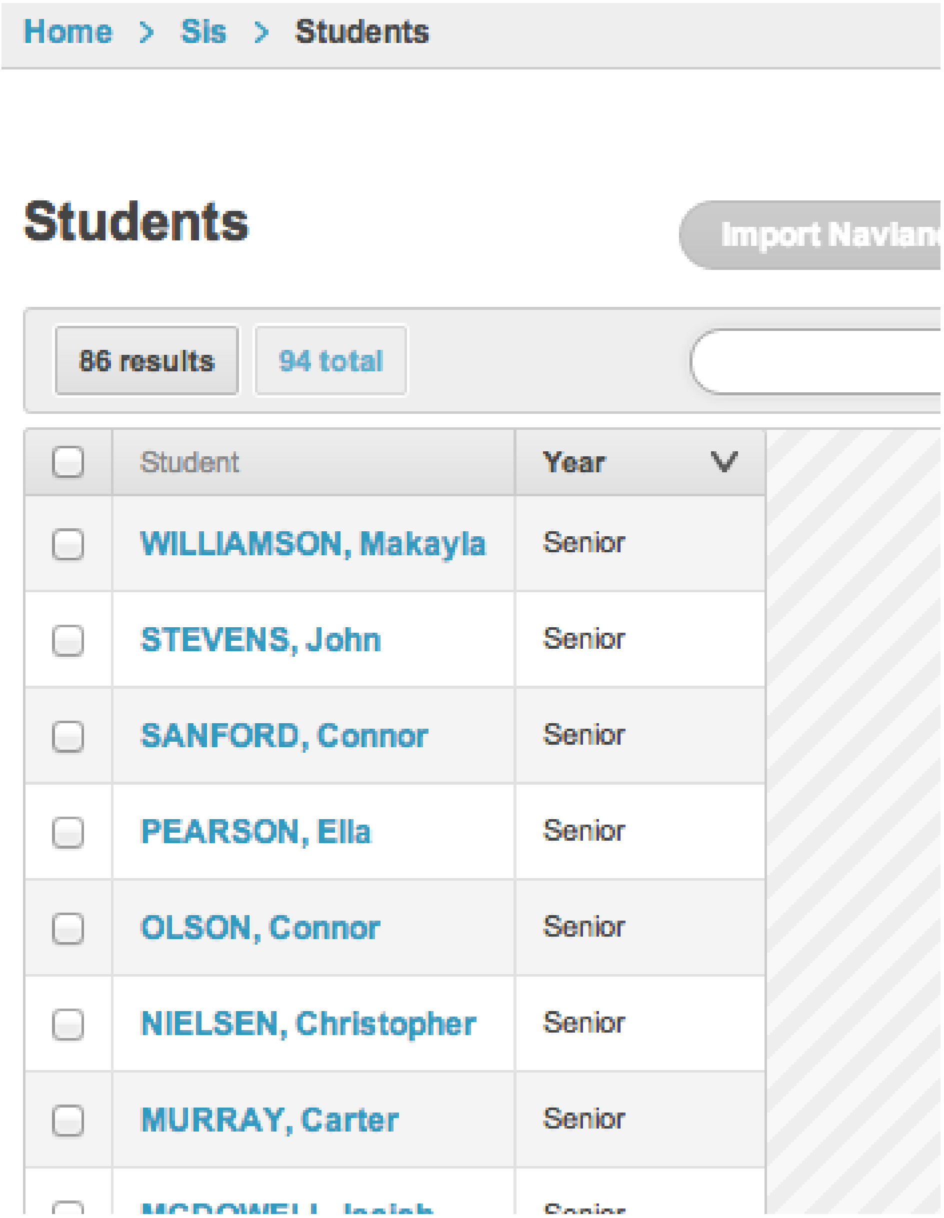

3. Enter the student's Last Name, First Name, and Username, which are required fields, and any additional information including Birth Date, Student Contact, and Notes. Click the **Save** button at the bottom right to complete the input of student information.




- Use the **Filter** function to filter students by Inactivity, Year classification, or Graduating Year.




- Click on the column heading **Year** to sort students by Year classification in ascending or descending order.




## 2.2 School Years

The starting, ending, and graduation dates of school years may be stored here. One year may be denoted as the active year, which may be used for calculations such as the number of discipline incidents.

## 2.3 Year Classifications

Year classifications are the various grades Django-SIS supports and their associated names. The defaults in Django-SIS are:

- Freshman: 9

- Sophomore: 10
- Junior: 11
- Senior: 12

## 2.4 Student Phone Numbers

In Django-SIS, you will see areas for Student phone numbers and for Student Contact phone numbers. To get proper information out of the system, you need to enter the numbers in the correct areas.

**Student phone numbers-** are for numbers to contact the student, for example a student's cell phone. If you use the home number in this spot, you should also put it into the contact area for the parent.

**Student Contact phone numbers**- cell, home, and work phone numbers for parents or other emergency contact personnel.

## 2.5 Student Contacts

The student contacts facet of django-sis refers to the parent or guardians associated with a student.

**Accessing the contacts**

There are two ways to add contacts. The first and easiest method is located on the student page in its own section titled, "Student Contact" respectively.

To add contacts, simply select a student from the Student > Edit page, or create a new student and scroll down to the *Student Contact* section and select **add**. Fill out the information in the dialog box and save.

The other method of accessing contacts works as follows:

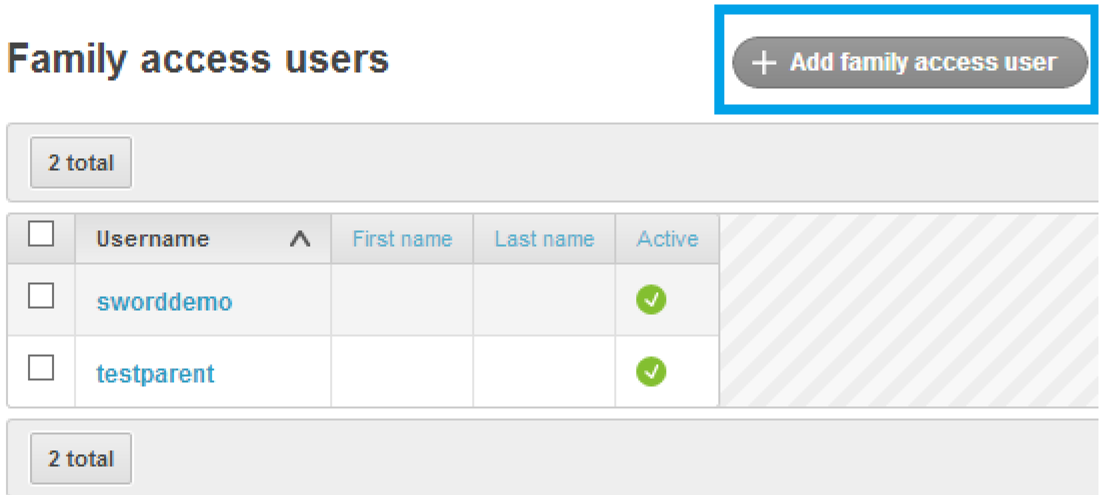
1. At the top right of your screen, select **View Site Admin**
2. Select **Student Contacts** from the *School Information* heading
3. Click **Add student Contact** at the top right, fill out available information, and save.

**What you can enter about each contact:** first name, last name, middle name, relationship to student, email, primary contact (checkbox), emergency only (checkbox), address, phone numbers, and student.

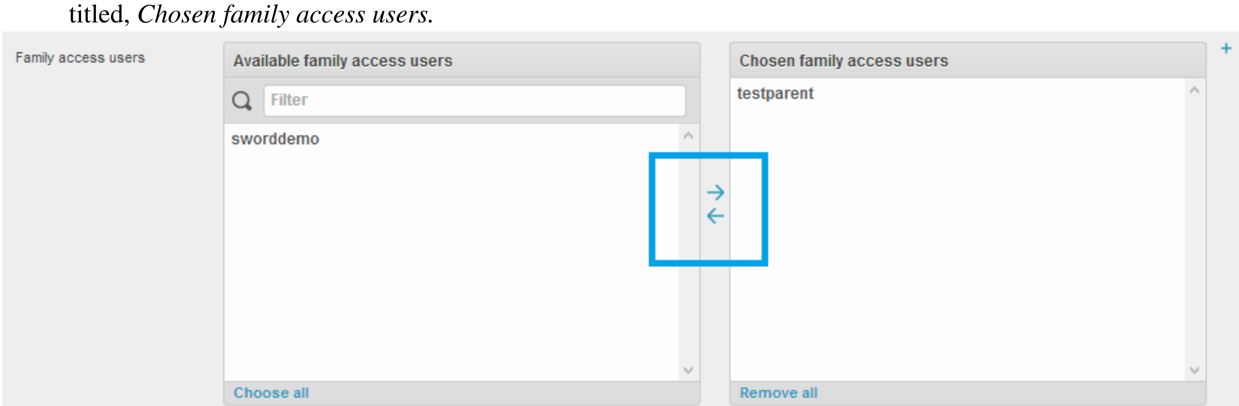
## 2.6 Family Access Users

Django-SIS allows for parents/guardians to log in and view grade information pertaining to their child. This section will show registrars or admins how to set up parent logins.

1. Under the **School Information** tab, select Add+ by the choice, **Family Access Users**




2. Create a username and password for the parent.
3. Return to the **Students** page, and under **Family Access Users**, select the user/s you've created that are in the left box titled *Available family access users* and click the arrow so that the user names switch to the right box titled, *Chosen family access users.*




4. Save

## 2.7 Cohorts

Cohorts are groupings of students within a school; the registrar may find this tool useful. For example, an "advanced class" cohort may be enrolled in particular classes, and homeroom placements may also be organized using cohorts.

## 2.8 Creating Courses

Creating courses in Django-SIS is a 3 step process:

1. Under the Courses and Grades tab, select +Add under **courses**


2. At the add course screen, enter any information that's appropiate- teacher, description, graded, etc. Keep in mind that the only **required** options to fill out are Fullname and Shortname.
3. Save

## 2.9 Enrolling Students

1. Under the Courses and Grades tab, select **courses**.
2. You'll be taken to a screen of all available courses, as shown below:

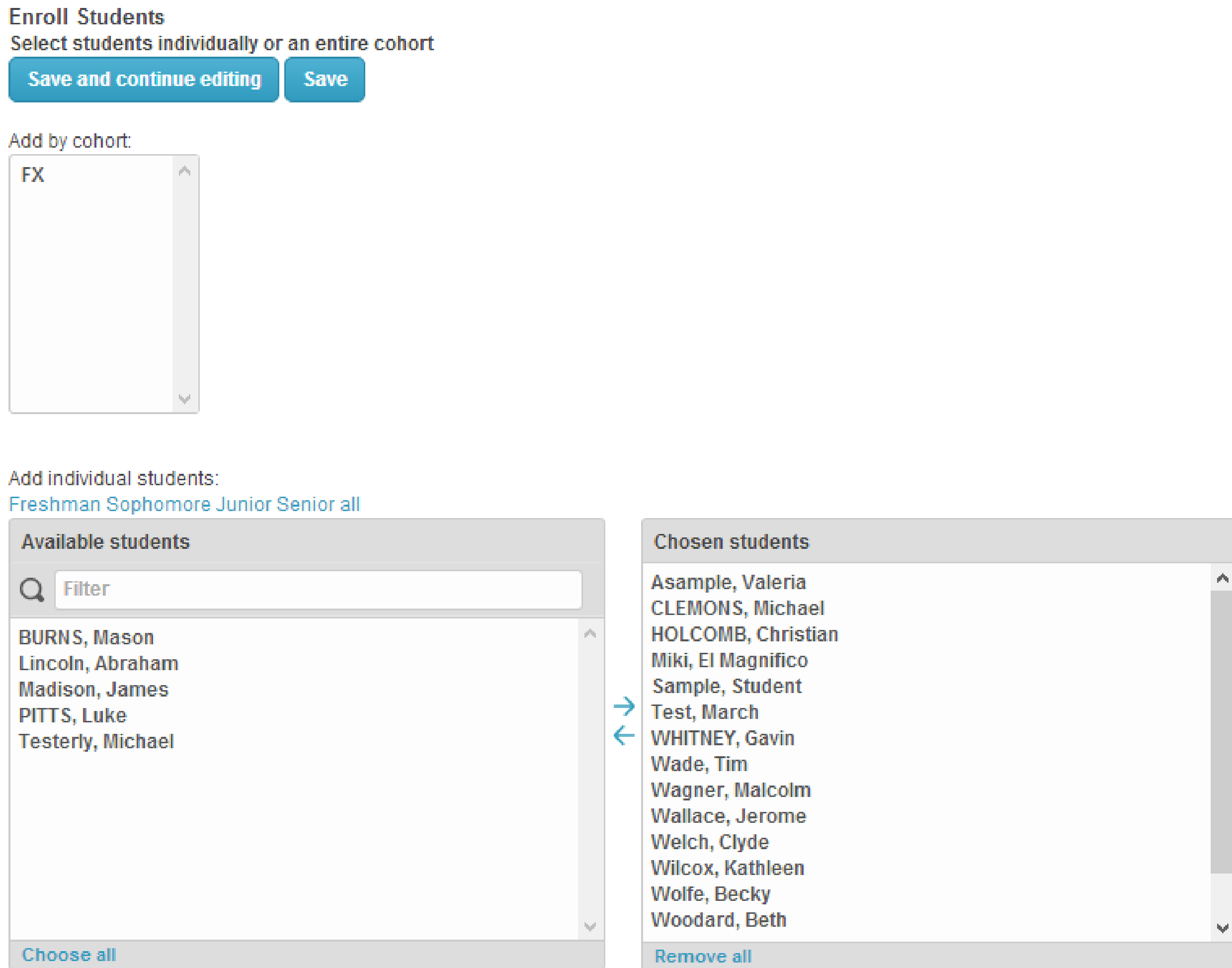
## Courses

9 total

| | Course | Teacher | Grades link |
|---|---|---|---|
| ☐ | English 101 | Old, Agnes | Grades |
| ☐ | Introduction to HTML | DemoTeacher, Demo | Grades |
| ☐ | Test | Old, Agnes | Grades |
| ☐ | Gradebook Example | , Tim | Grades |
| ☐ | Blargity Derping | Teacher, Amy | Grades |
| ☐ | Introduction to Python | DemoTeacher, Demo | Grades |
| ☐ | History of Technology | DemoTeacher, Demo | Grades |
| ☐ | That other homeroom | Old, Agnes | Grades |
| ☐ | Some homeroom 2012 | Teacher, Amy | Grades |

9 total

3. Select the course you will be enrolling students in.
4. Once selected, you will be taken to the change course screen. Click the **Enroll Students** button located towards the top right of the screen.
5. The screen below appears- move student/s from the available students box on the left, into the *chosen students* box on the right. You may filter the available students by grade level, located directly above the *available students* box.




6. Save

## 2.10 Submitting Grades

Teachers may submit grades using three methods: manually, import via spreadsheet, or using Engrade.

**IMPORTANT:** Django-SIS only stores final grades.

**Manual Method**

The manual method of entering final grades works as follows:

1. Under the Courses and Grades tab > Courses, click on the appropiate course.
2. Once selected, click **Grades** on the course page, located towards the top left.

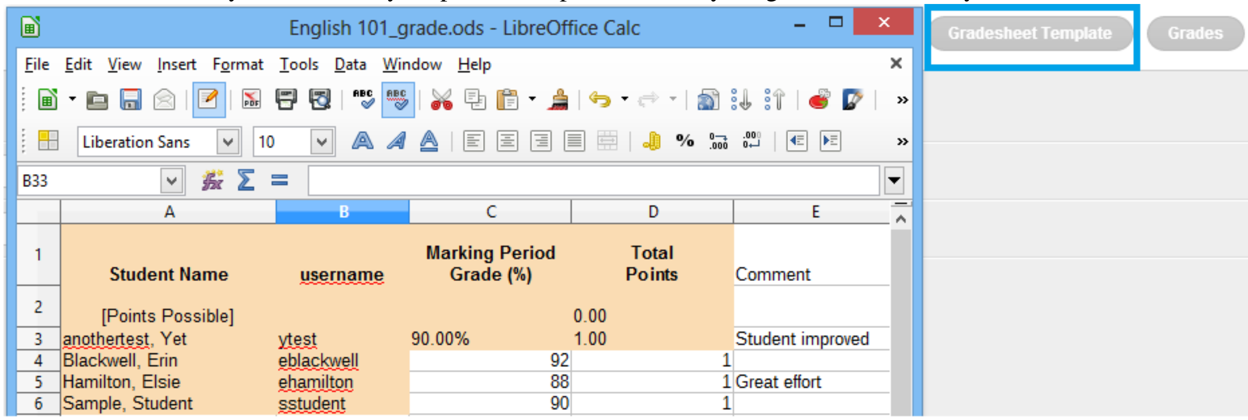
3. You'll be taken to a screen as shown below. Click on the cell and type a grade, then save. .. image:: /images/gradesmanualentry.png

**Spreadsheet Method**

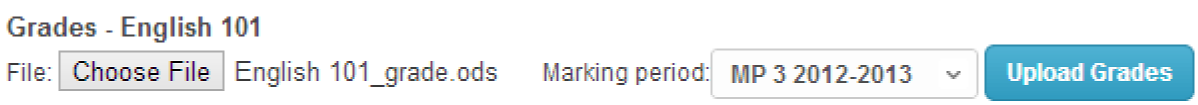
NOTE: Only final grades are stored in Django-SIS. Assignments should be stored in a Spreadsheet based on a template your administrator created.

Access the course page, similar to step 1 from above.

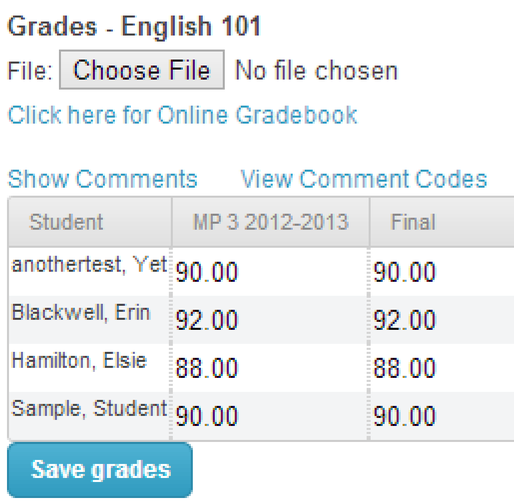
1. Under the Courses and Grades tab > Courses, click on the appropiate course.
2. Select **Gradesheet Template**. It will open up an Excel file already set up for grade input with usernames and students already available for you, per the template. Fill out your grades and note any comments.

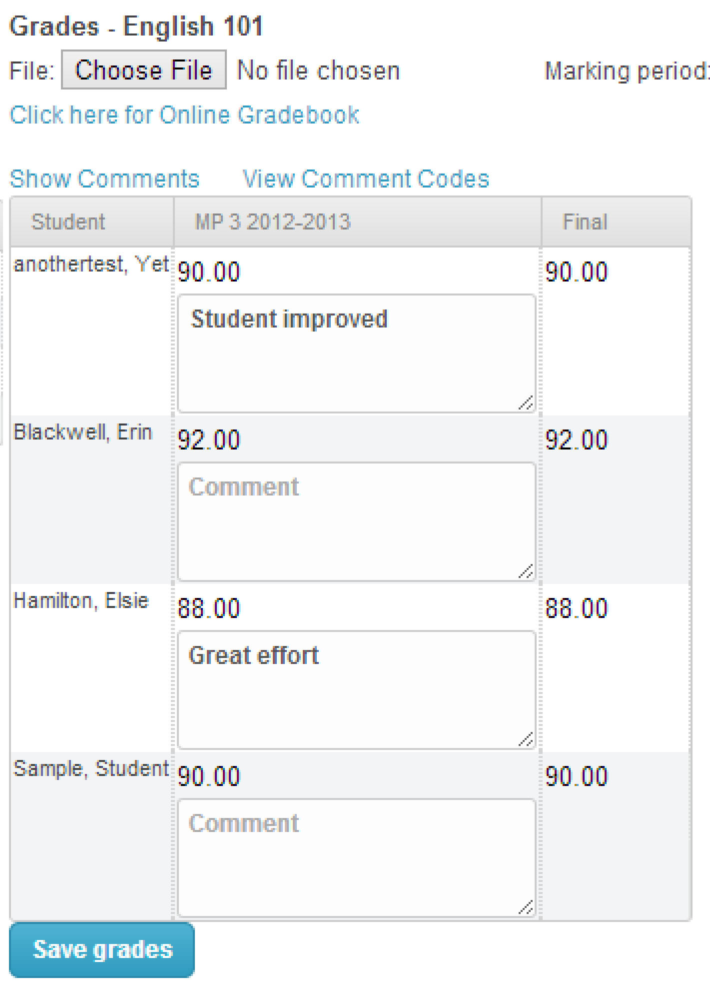



3. Once the template is filled out. Select **Grades** from the course page.
4. *Ensure* that the marking period is correct and matching the **tab name** on the spreadsheet you filled out. Select **Upload**




5. Django-SIS will then set the grades and comments in accordance with your spreadsheet.




- Final grades are calculated automatically, but may be overridden by privileged users.
- Mid marking period grades will never effect any calculation.

*Grade Comments/Comment Codes:*

- Comments may be entered via plain text or comment codes as designated by your school. If you wish to enter multiple comment codes per student, a comma separating each code is necessary (i.e.14, 3).
- Blank comments or comment codes will be ignored.
- If a comment or code already exists and you want to delete it, select the code and replace it with "none".

*Tips:*

- Django-SIS stores only two decimals, although calculations may be done with more.
- You may enter approved Letter grades if desired such as P and F. These will not effect calculations.
- You may only be allowed to change specific marking period grades determined by your administrator. If you've made a mistake you may need to contact an administrator or registrar. It is possible to grant you access to directly enter grades in Django-SIS. This may be useful for Pass/Fail grades.

**Engrade Method**

Django-SIS is able to sync with the online gradebook, Engrade so teachers my store all assignments there, then sync final grades there to appear in Django-SIS.

1. Under the Grades menu item at the top, select **Submit Grades**
2. This will reveal your list of courses, along with the option to download a blank gradebook. Below these options is the button **Sync all grades from Engrade**.

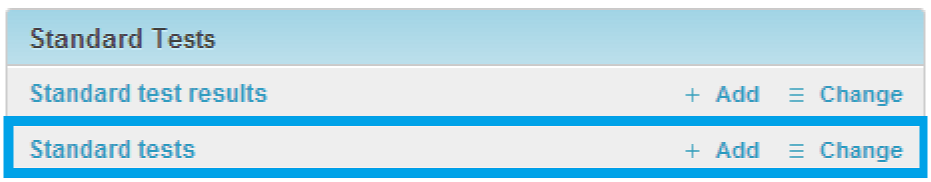
**Keep in mind that all grades synched from Engrade will override any grades that have been entered into Django-SIS for that marking period.**

## 2.11 Standard Tests

Django-SIS allows users to create and enter information for various standard tests (ACT, SAT, PLAN, etc.).

**Creating a New Standard Test**

1. Under the *Standard Tests* tab, select +Add by the **Standard tests** option.

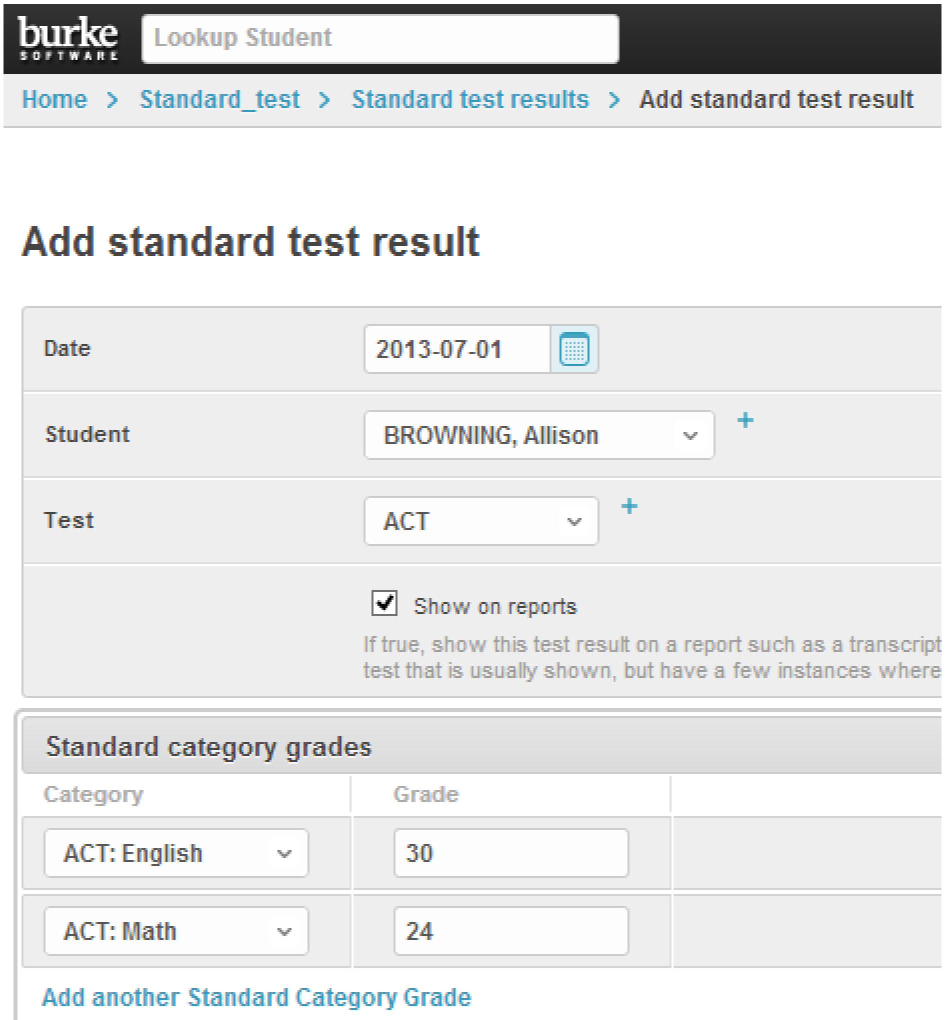



2. Enter a name of the test and select the appropiate boxes on how to calculate test results.
3. The **Standard Category** tab refers to the categories of a test (English, Math, Science, Writing, etc.)
4. Save

**Recording Standard Test Scores**

1. Under the *Standard Tests* tab, select +Add by the **Standard test results** option.
2. Enter a date, select the student, and select the appropiate test.
3. Under **Standard category grades** select the cateogry and type in the grades.

**Note** You add additional categories by selecting *Add another Standard Category Grade (Math, then Science, etc.)

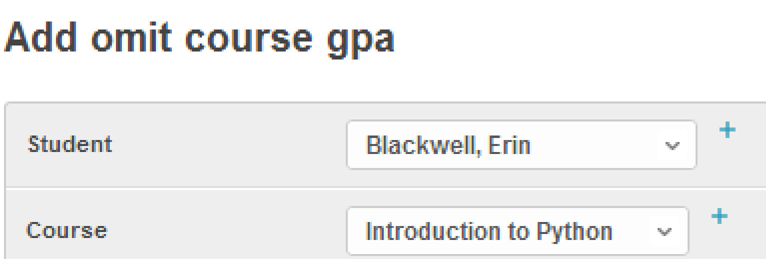



4. Save

## 2.12 Omit Course GPA

Django-SIS allows users to quickly omit a student's grades for a particular course.

1. Under the Courses and Grades tab, select +Add by the **omit course GPA** selection

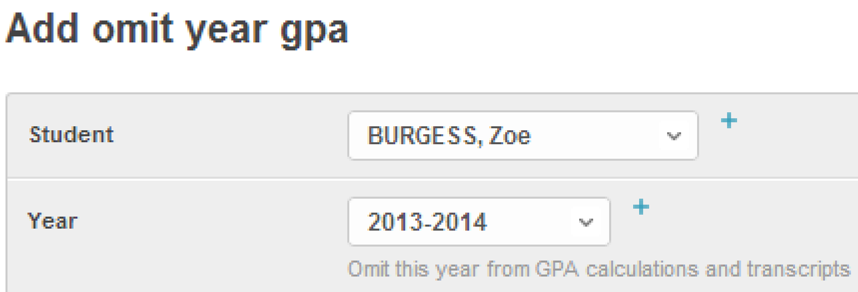



2. Select the appropiate student and corresponding course.
3. Save

## 2.13 Omit Year GPA

Similar to omitting course GPAs, Django-SIS allows users to omit an entire year of grades for a student so that particular year will not be calculated into GPAs and transcripts. The process is similar to omitting for a course, as shown above.

1. Under the Courses and Grades tab, select +Add by the **omit year GPA** selection




2. Select the appropiate student and corresponding year.

3. Save

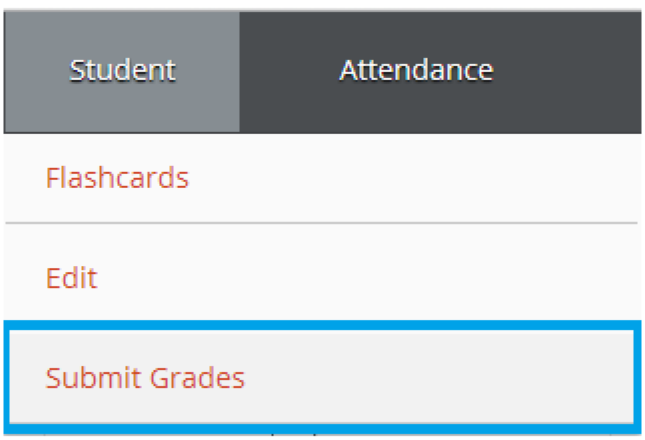
# Django-SIS Gradebook

Django-SIS comes packaged with a gradebook for teachers to utilize to store and track grades. This section will discuss and review how to use and configure the gradebook for your needs.

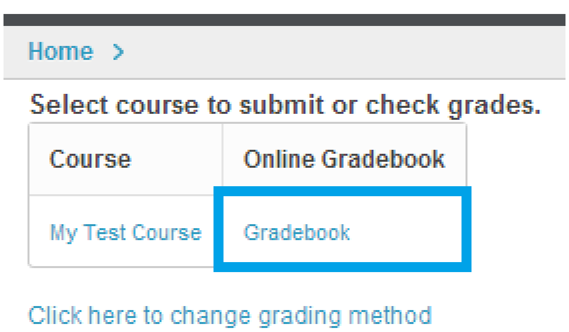
## 3.1 Accessing the Gradebook

To access your gradebook to submit grades, do the following:

1. Log in to Django-SIS
2. Select **Submit Grades** from the *Student* drop down at the top menu.




3. Django-SIS will show you a list of your courses. Click on **Gradebook** by the appropriate course.

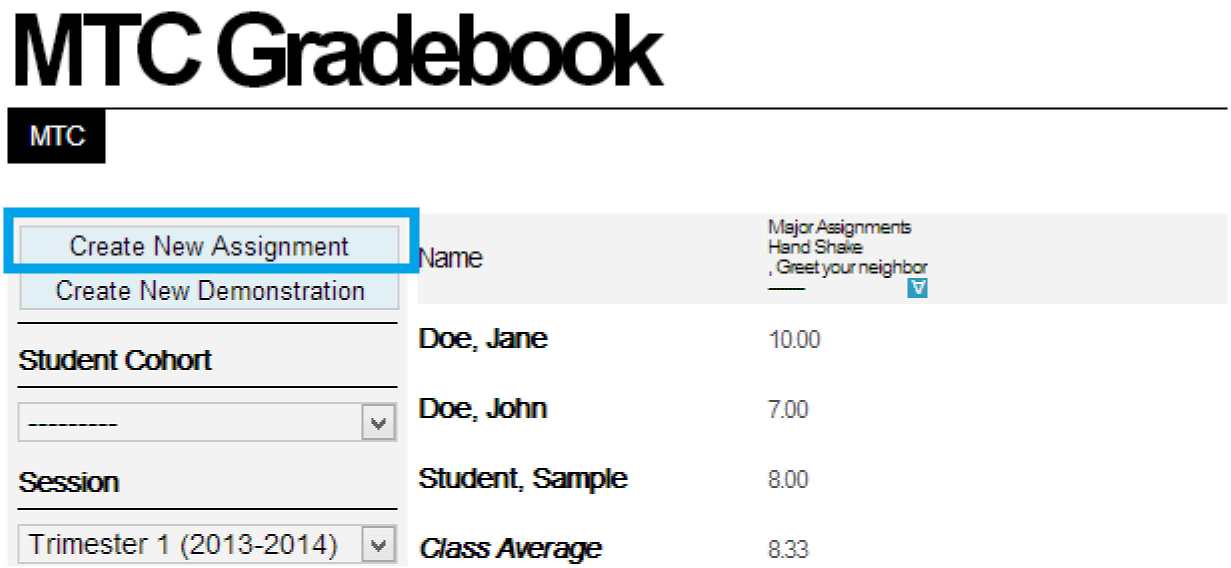



4. The gradebook for the selected course will then open up.

## 3.2 Creating Assignments

After you've accessed your gradebook, and you're ready to create assignments:

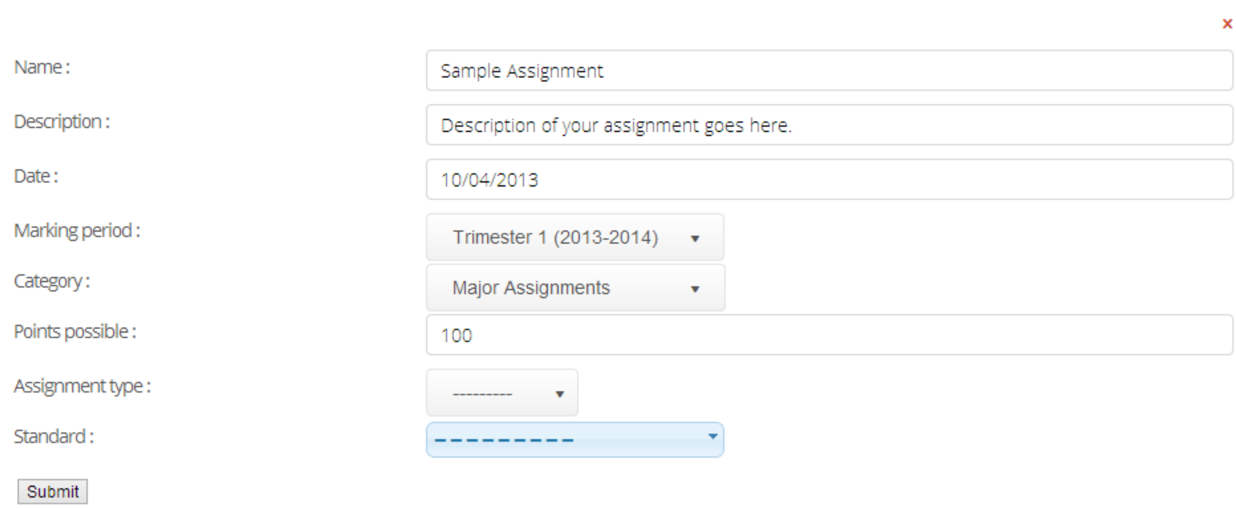
1. Select **Create New Assignment** from the gradebook.




2. Once selected, a dialog box will pop up asking you to enter information about your assignment. Enter your information and submit.

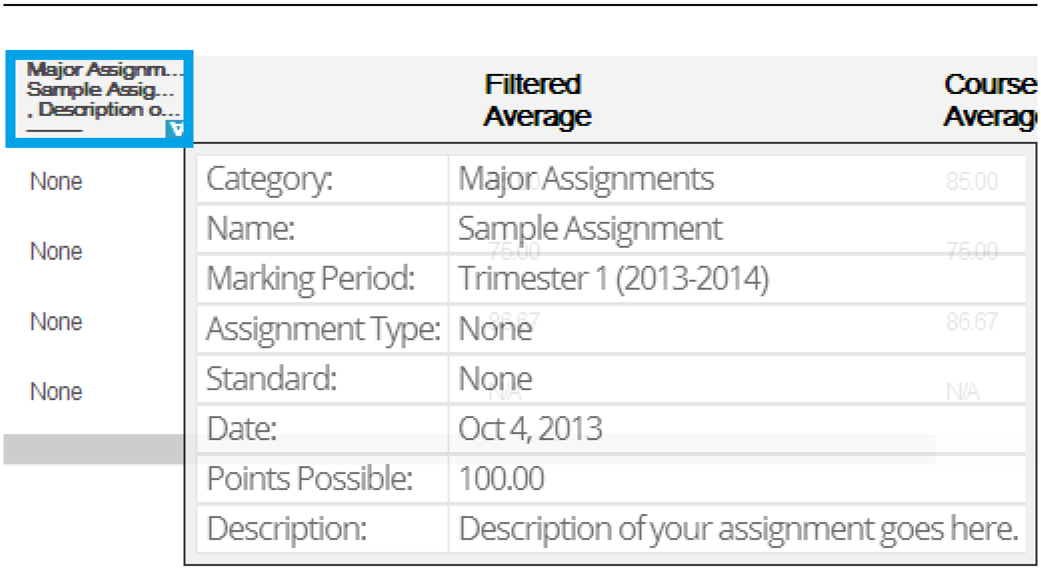
**Note:** You'll want to ensure that you enter a **category** for weighting purposes. *Standard* and *Assignment Type* here are optional and used for benchmark grading purposes, and might not apply to you (the options will be blank if not applicable for you).

See the image below:




**Note:** Django-SIS will calculate grades based off the number you enter for **points possible**. For example, if your assignment is one where 10 points is the max, and a student gets a 7, that will accordingly effect his overall average as if it were a 70.

3. Your newly created assignment will now be listed as a column header. Hovering your mouse over it shows the details of it. See image.



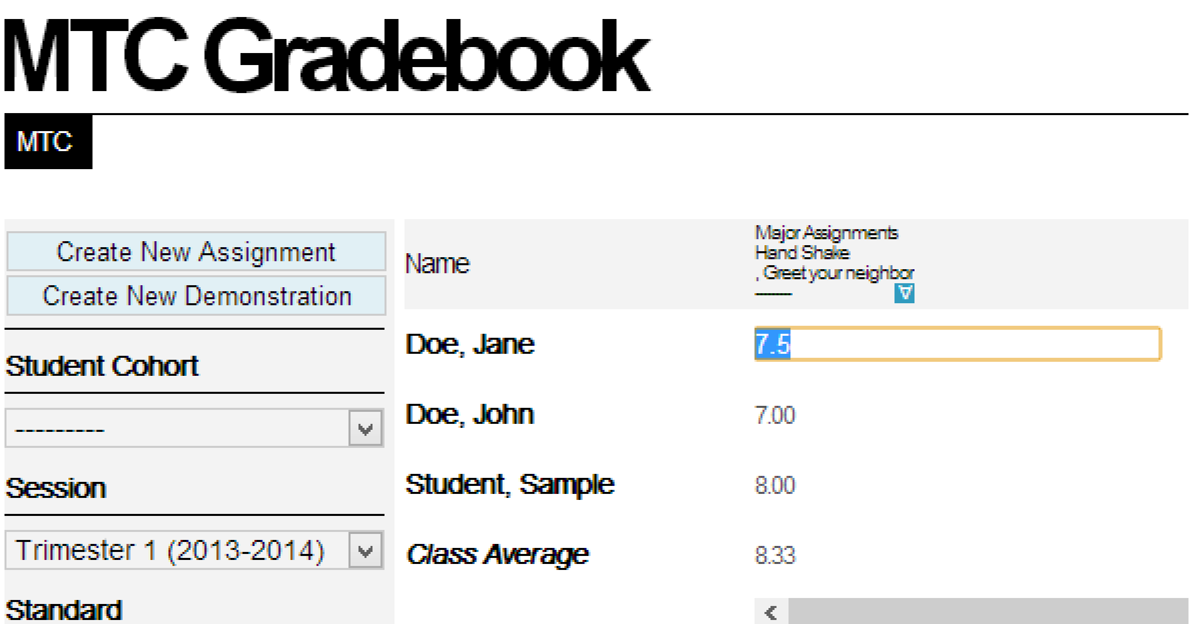

## 3.3 Entering Grades

Entering grades for assignments is a straightforward process and can be done in two ways:

1.) **Manually enter grades for each student**

-This process involves clicking on the appropriate cell by the student, and typing in a grade. After you type in a grade, press Enter or click on the next cell.

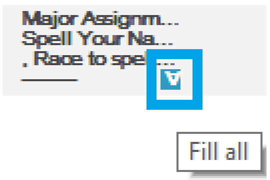
The example below shows an instance of giving student *Jane Doe* a grade of 7.5 for the **Greet your neighbor** assignment that is graded out of 10.



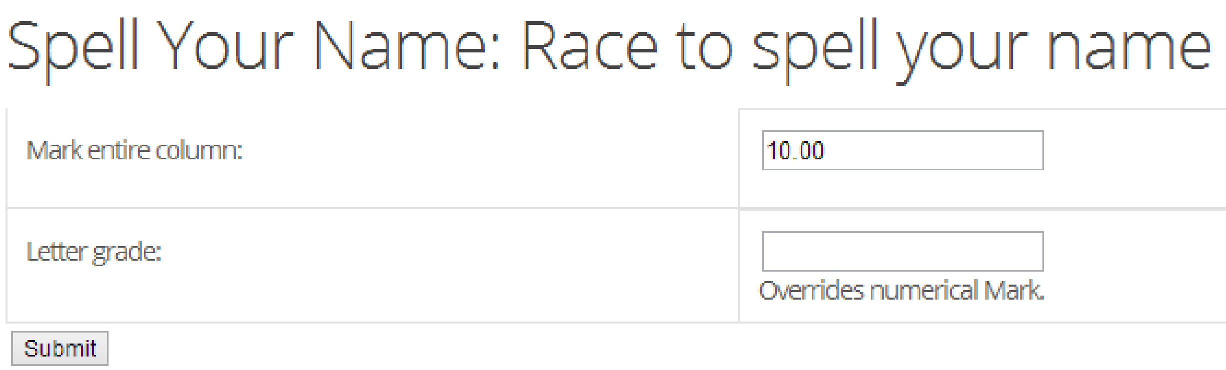

2.) **Using the fill-all feature**

-This feature can be used on assignments, and will fill all student grades with whatever you enter. Accordingly, it's best used when the majority of students received the same grades where you then change the few unique grades.

How to: click the little blue symbol by the assignment.




A dialog box will pop up with the assignment name asking you what grade all students should receive:

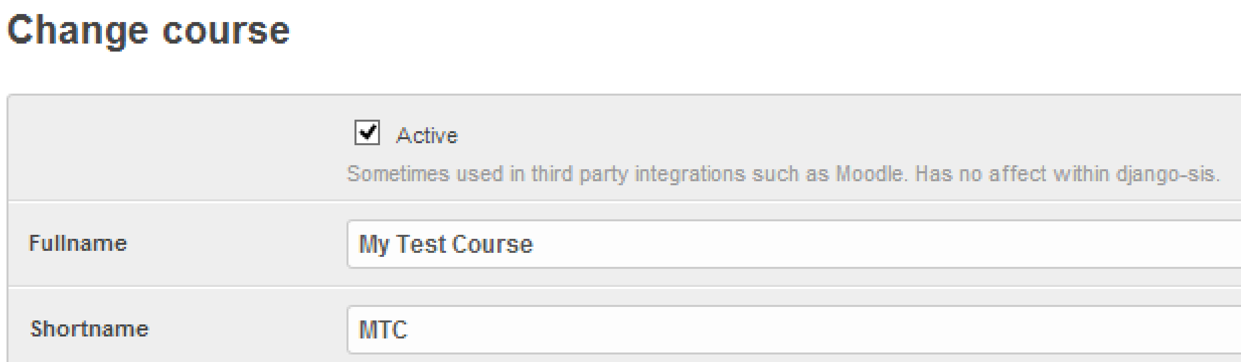



Enter a value and **submit**.

## 3.4 Other Notes for Teachers

-The title of your course gradebook that you see at the top is whatever your registrar has entered as the **shortname** for the course. This can be changed. Email your registrar who created the course and ask for a change if you need one.

In this example, the full title of the course is "My Test Course", however, on the gradebook I see "MTC" Gradebook, as the registrar had "MTC" as the **shortname**.




-**Course averages** for students will calculate automatically after you enter a grade. *Filtered Average* won't update until the page is refreshed or the filter is changed.

-**Class average** will also calculate automatically.

-Always enter a category for assignments so Django-SIS knows how to calculate averages.

-**Course average** column is what students will see as their grade for the course on report cards.

## 3.5 Grade Calculation Rules

**Note** This section is mostly intended for registrars, as it involves special permissions to access and edit. Calculation rules must be set prior to utilizing the django-sis gradebook, so the gradebook can know how to calculate averages correctly.

To access grade calculation rules:

1.) Click on *View Site Admin* at the top right of django-sis.

2.) Under the *Courses and Grades* heading, select *Calculation rules*

3.) Click add rule if a calculation rule does not exist.

**Note:** Once you set up a calculation rule, it will automatically apply to every subsequent year, unless you create a new, more recent rule. They do not need to be set every year, unless the calculation rules at your school change.

There are three setups required at the calculation rule page: first year effective, points possible, and decimal places.

**First year effective** - the school year to first apply the grade rules to.

**Points possible** - number field. Maximum amount of points for each grade book entry. 100 is typical.

**Decimal places** - number field. How many decimal places should be included for each gradebook entry.

The section below the above requirements is titled, *Categories included in each course average*. This is where the registrar sets up how categories will be weighted. There are three columns in this section: category, weight, and apply to departments.

**Category** - categories are types of assignments teachers may enter grades for. Ex. Quizzes, Major Assignments, Character Counts, Tests. Teachers are **required** to enter a category type when they are entering a new assignment in the gradebook to grade.

**Weight** - how to weigh each category. ex. if you have 4 categories, you can break them up equally by assignining a weight of 0.25 for each.

**Apply to departments** - which apartments these categories and weights effect.

Be sure to save after assigning categories, weights and departments.

## 3.6 How Course Grades are Calculated

This section is intended for any staff interested in how django-sis calculates grades.

Final grades may be calculated in two ways. First, the software will look to see if any final grades have been overrided. In other words, if a registrar manually alters the final grade on a gradebook, that new grade will remain.

If there are no overrides, the final grade for a course will be the weighted average of the number grades accordingly.

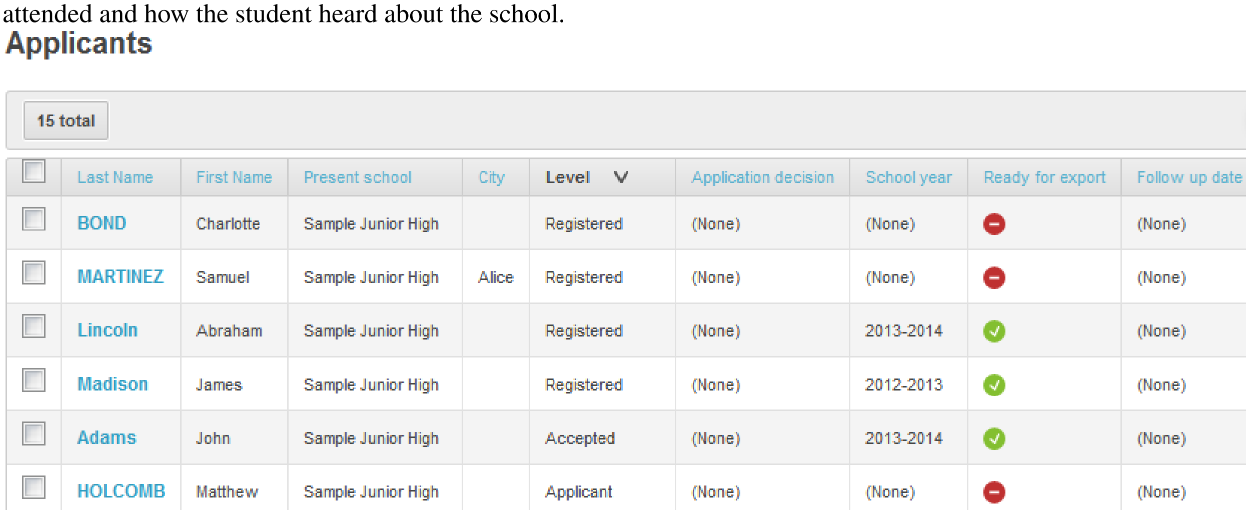
If a particular assignment entry is a Pass or Fail (indicated by a P or F, respectively) then the gradebook will initiate a special letter grade calculation where P = 100%, and F = 0%. Since the gradebook will take averages, 2 P's and 1 F will result in an average of 66.66%. Keep in mind the teacher will always just see a P or F.

Some calculations may have dates attached for historical purposes or different marking period weights throughout the year.

CHAPTER 4

# Admissions

The admissions module allows schools to keep track of applicants, and their status in the application process. Each step in the application process can be customized to fit a school's unique need. Users can designate steps that need to be completed before moving onto the next level. Additionally, Django-SIS may track any open houses a student has attended and how the student heard about the school.


Applicants

15 total

| | Last Name | First Name | Present school | City | Level | Application decision | School year | Ready for export | Follow up date |
|---|---|---|---|---|---|---|---|---|---|
| | BOND | Charlotte | Sample Junior High | | Registered | (None) | (None) | | (None) |
| | MARTINEZ | Samuel | Sample Junior High | Alice | Registered | (None) | (None) | | (None) |
| | Lincoln | Abraham | Sample Junior High | | Registered | (None) | 2013-2014 | | (None) |
| | Madison | James | Sample Junior High | | Registered | (None) | 2012-2013 | | (None) |
| | Adams | John | Sample Junior High | | Accepted | (None) | 2013-2014 | | (None) |
| | HOLCOMB | Matthew | Sample Junior High | | Applicant | (None) | (None) | | (None) |

The image above details the dashboard that an admissions counselor or designated user sees when the admissions module is selected. Most modules include a dashboard to provide users a general overview of information that is able to be filtered.

## 4.1 Adding an Applicant

To add an applicant:

1. Select **Applicants** under the Admissions module.
2. Enter information about the applicant accordingly. First and Last Name fields are required.
3. Click **Save**.

Django-SIS will then return you to the applicant's dashboard where you will see your newly-created applicant at the top.

## 4.2 Admissions Levels

Django-SIS allows schools to control admissions levels/steps that are unique to their process. Each step is customizable as follows:

1. Select **Admissions Levels** under the Admissions module.
2. You will see the screen shown below.

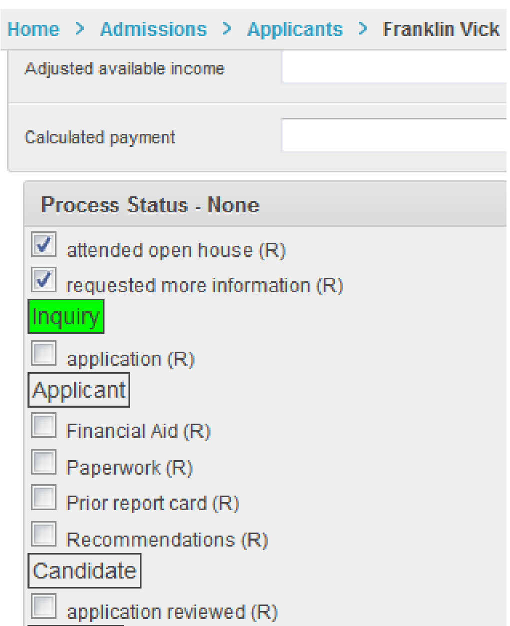
### Admission levels

| | Edit | Name | Order | Show checks |
|---|---|---|---|---|
| ☐ | Edit | Inquiry | 1 | \|attended open house \| requested more information \| |
| ☐ | Edit | Applicant | 2 | \|application \| |
| ☐ | Edit | Candidate | 3 | \|Financial Aid \| Paperwork \| Prior report card \| Recommendations \| |
| ☐ | Edit | Decision | 4 | \|application reviewed \| |
| ☐ | Edit | Accepted | 5 | \|accepted \| |
| ☐ | Edit | Registered | 6 | \|Deposit Paid \| |

6 total

3. From this screen you can add an admissions level, selecting the **Add Amissions Level** button or edit an existing one by selecting *edit* located next the level you are altering. From the edit screen or add screen, make the necessary changes/additions and then select save.

The section under the header, **Items needed to be completed to attain this level in the process**, refers to creating a checklist of various tasks the applicant needs to complete prior to reaching a new step. For example, the image below details a checklist containing the two required tasks 'Open House' and 'Request more information' which must be completed before the applicant reaches the level of Inquiry.




Users may designate levels required in order to advance. For example, schools may require an applicant pay an initial deposit prior to registration. To make a step required, simply check the box found under the **Required** column and save.

## 4.3 Filtering Applicants

To maximize organization, efficiency, and promote the ease of collecting various admissions data for report preparation, Django-SIS contains several filters and functions accessible through the main applicant page. Each column header in the image below will sort accordingly. For example, clicking on Last Name will filter by last name, application decision by decision, etc.

| | Last Name | First Name | Present school | City | Level | Application decision | School year | Ready for export | Follow up date |
|---|---|---|---|---|---|---|---|---|---|
| | Adams | John | Sample Junior High | | Accepted | (None) | 2013-2014 | | (None) |
| | BOND | Charlotte | Sample Junior High | | Registered | (None) | (None) | | (None) |
| | Carter | Jimmy | Sample Junior High | | Inquiry | (None) | 2013-2014 | | (None) |
| | Clinton | Bill | Sample Junior High | | Applicant | (None) | 2013-2014 | | (None) |

Alternatively, users may choose from the available filters located directly to the right of the applicant list. The drop down list allows users to select and combine the following filters: school year, level, checklist, ready for export, present school, ethnicity, heard about us, and year. The filter tool will do so in real time, no need to select and save.

## 4.4 Exporting Applicants

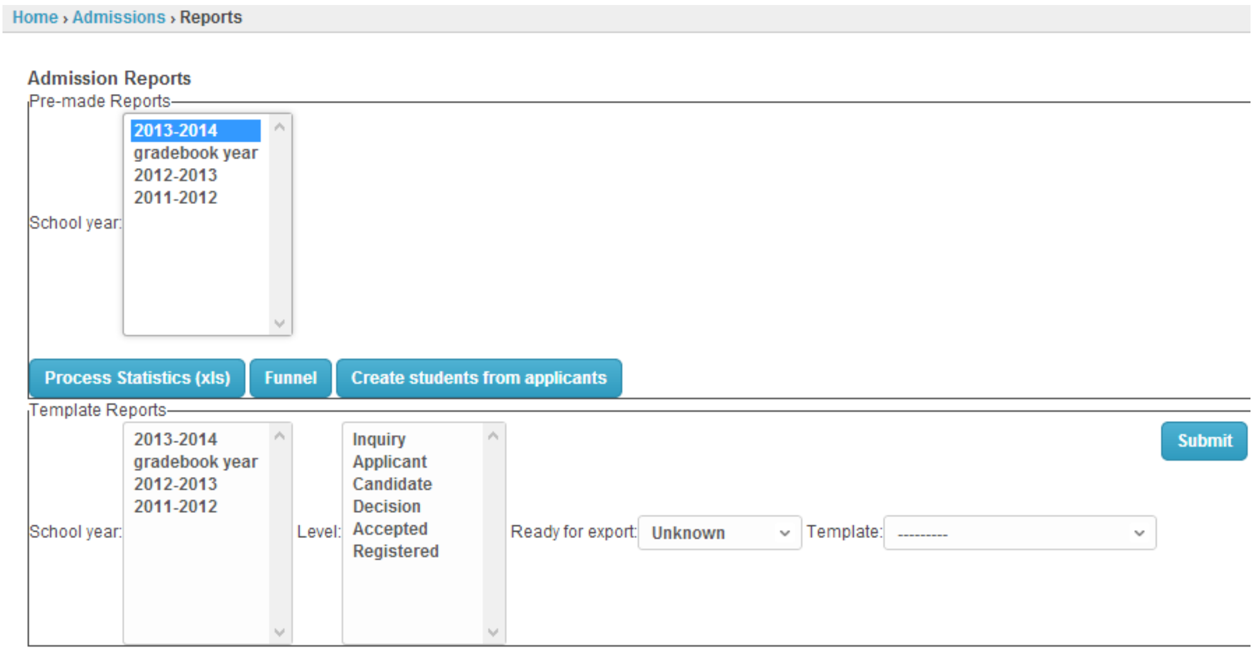
Django-SIS allows for easy export into an Excel document for sharing or distribution. After applying filters to applicants, follow the steps below to export into an Excel file.

1. Select each applicant you would like to export or select all by selecting the top left checkbox.
2. Click the drop down menu located on the black toolbar at the bottom of the page.
3. Select **Export to XLS**. A box opens up with options on what to export.
4. Choose Select All to export all information entered for each applicant or check specific boxes.
5. Scroll down and select **Submit**.
6. Django-SIS will then open an Excel document.

## 4.5 Admission Reports

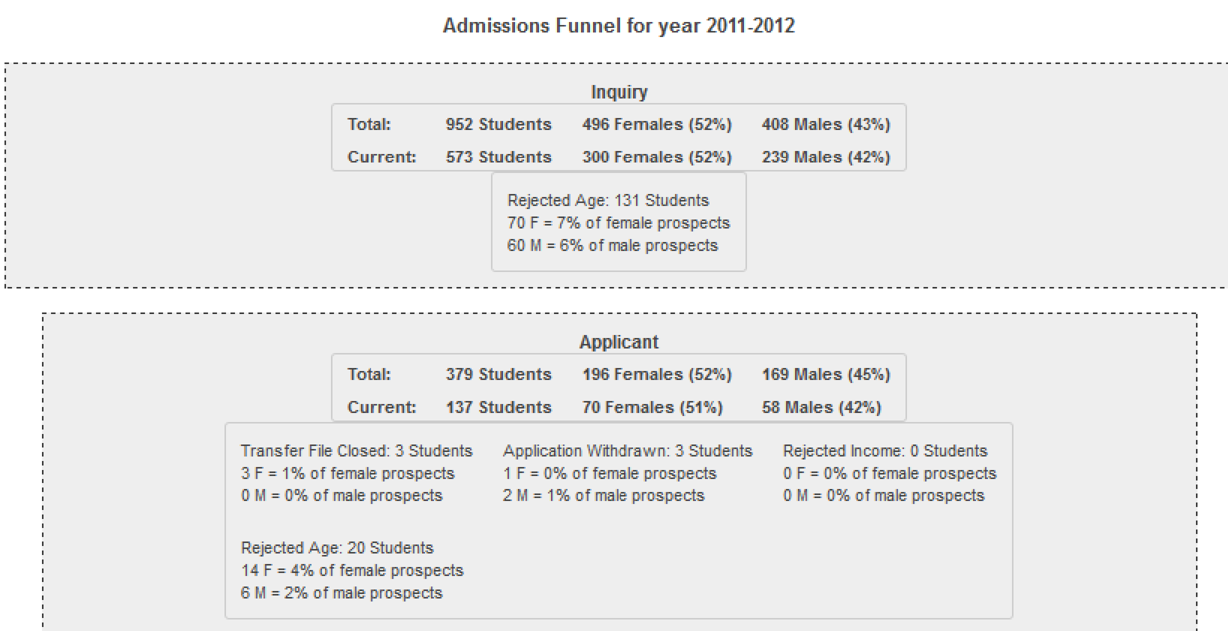
Some basic Admission Reports are available built in to Django-SIS that allows users to quickly process statistics based on a school year's applicants.

1. Under the **Admissions** tab in the navigation bar, select **Reports**




2. Select a year and click **Process Statistics**.
3. Django-SIS will generate an Excel document detailing some basic admission statistics such as number of applicants by grade or number of applicants on a particular level in the process.

In step 2, another option is to choose **Funnel**, which generates on-screen admissions statistics from each step in the admissions process. The report shows total, current, male/female, and rejected reasons.




## 4.6 Creating Students from Applicants

Prior to beginning a new school year, a school will eventually need to convert the applicants into enrolled students to assign classes, grades, etc.

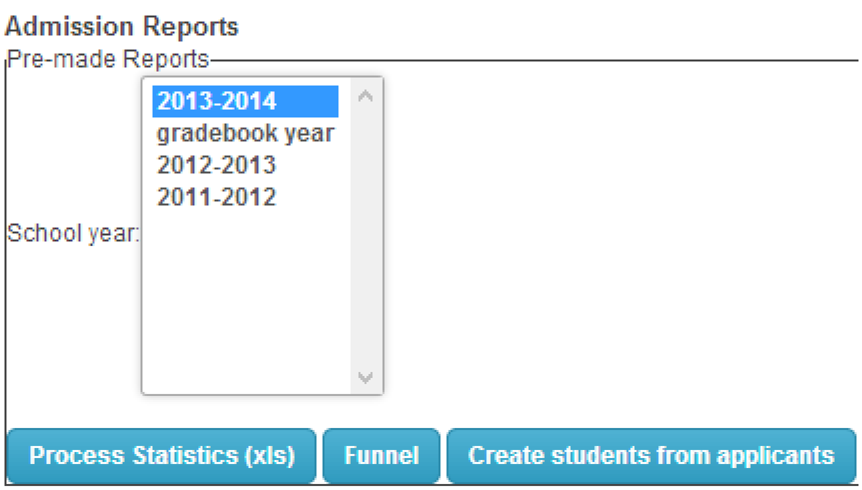
**IMPORTANT NOTE:** It is important to keep in mind that the only applicants who will be made into students, are those applicants that have the **Ready for Export** check by their name on the dash. Accordingly, marking students as ready for export should be the absolute final step in the process.

| ☐ | Last Name | First Name | Present school | City | Level | Application decision | School year | Ready for export | From online inquiry | Follow up date |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | Testerly | Michael | (None) | | Candidate | (None) | 2013-2014 | ✔ | ⛔ | (None) |
| ☐ | Monroe | James | (None) | | (None) | (None) | (None) | ✔ | ⛔ | (None) |
| ☐ | Gallagher | Claire | Sample Junior High | Austin | Applicant | Wait Listed | 2013-2014 | ⛔ | ⛔ | (None) |
| ☐ | Vick | Franklin | hundred acre wood | | (None) | Wait Listed | 2013-2014 | ⛔ | ⛔ | (None) |

In the example above, only Michael Testerly and James Monroe will be made into students.

To create students from applicants, follow the steps below:

1. Access the Admissions report screen, by selecting Admissions > Reports from the menu located at the top of your page.
2. Select the appropiate school year.
3. Click the **Create students from applicants** button.




## 4.7 Modifying options - Admission Administration

The remaining selections found under Admission Administration such as feeder schools, ethnicity choices, religion choices, school types, etc., allow the dropdown menu choices to be modified. For example, if a particular religion choice is unavailable in dropdown, click on Religion Choices under Admission administration, then the +Add religion choice button to enter the religion, then Save. The entry is now permanently available in the dropdown menu.

CHAPTER 5

# Attendance

Django-SIS has a built-in attendance module that allows teachers to record daily attendance. Homerooms must already be in place, which are simply courses that are designated as such.

## 5.1 Taking Attendance

1. Click **Attendance** from the navigation menu.
2. A screen appears with a class list. Teachers can mark all students present by **Set all to Present** or click the dropdown menu to mark individually.

Additional comments may be entered in the **Notes** column.

**Things to keep in mind:**

- If a student is already marked absent before, the teacher will see this. At this point, nothing the teacher does will affect it. Keeping it as absent will not change it. Marking present will also not change anything.
- Teachers are not allowed to edit attendance records.
- If a student is enrolled in two different homerooms and is marked absent in one and present in the other, the student will be considered absent.

## 5.2 Attendance Reports

Under **Attendance**(navigation menu) and **Reports** are a number of pre-formatted attendance reports designed to be quickly exported into an Excel or Word document. The available reports are:

**Daily Attendance** This report allows users with permission to generate the daily attendance for all students, separated by grade. In particular, the report displays all *absent* students (not marked Present), reasons, and year classifications. Total absences by year classification are tallied at the bottom.

**Lookup Student** Allows users to look up a student's attendance record. The date/reasons for all absent/tardy/late excused, etc. are reported in a Word document.

**Perfect Attendance Certificates** For a date range or year, this report generates a Word document with a list of students who have zero absenses and tardies.

**Daily Attendance Stats** For a date range or marking periods, this report generates an Excel document showing the date, number present, number absent, and absent percentage.

**By Student Report** This report generates an Excel document of every enrolled student, displaying a tally of all absences and tardies including type of absence (excused, medical, holiday, religious, etc.).

**Aggregate Report** For a date range or marking period, this report is a combined tally of all absences. An absolute Absent Percentage is also reported.

## 5.3 Editing Attendance

Users with the proper permissions may be allowed to edit attendance for the entire school. To do so,

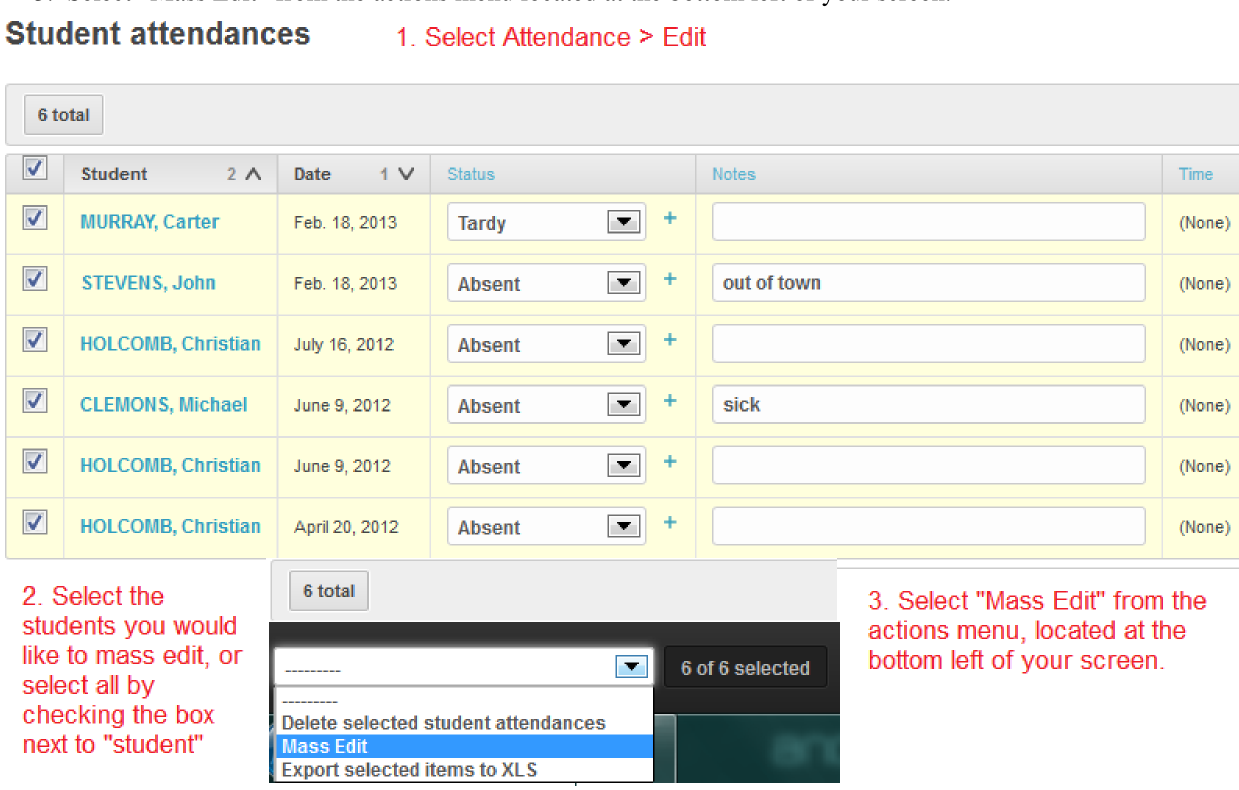
1. Select **Attendance** from the navigation menu, then **Edit**.
2. The Edit screen will display all students who have *not* been marked *Present*; edits/notes may be entered. For example, if the school later receives a doctor's note for an absent student, *Absent* may be switched to *Absent Excused* with a *Doctor Visit* note.

This dashboard also contains a filter option located to the right of the screen, allowing filtering by date, date range, or attendance status (absent, tardy, absent excused, etc.).

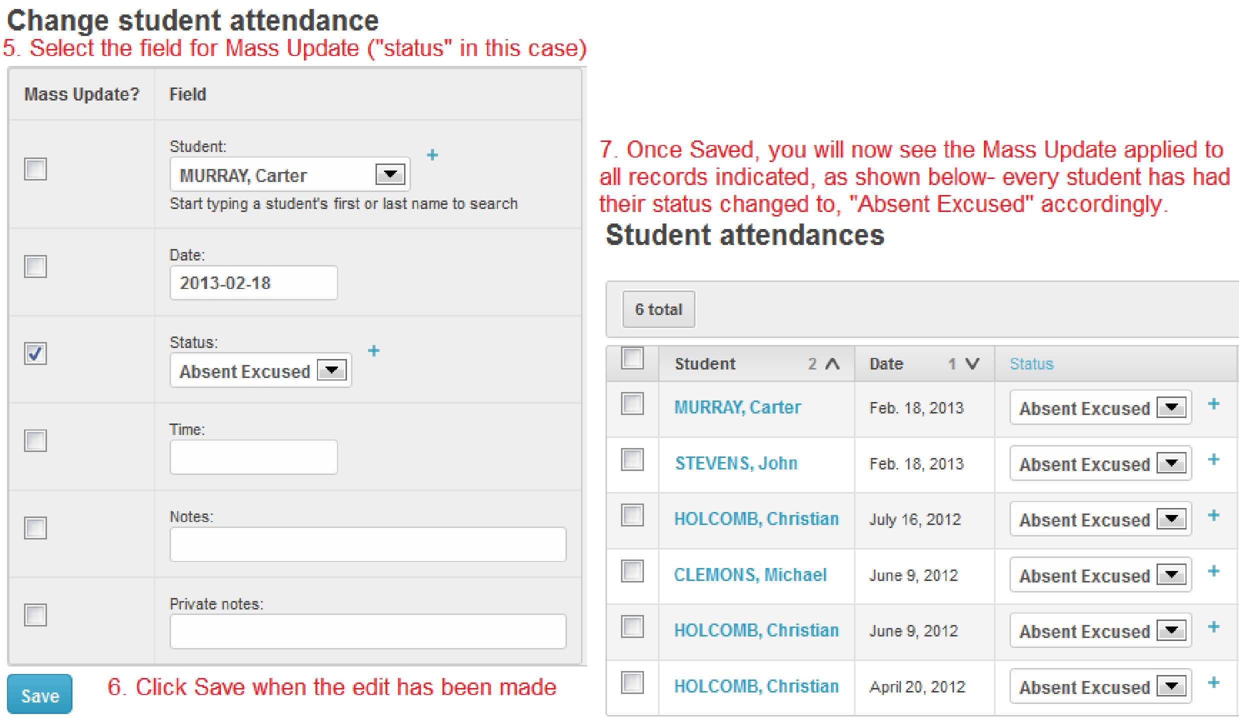
## 5.4 Mass Edit Attendance

Django-SIS allows users to edit multiple records at once, using the mass edit tool as detailed and shown:

1. Select Attendance > Edit
2. Select students for edit
3. Select "Mass Edit" from the actions menu located at the bottom left of your screen.



4. The next screen will allow fields to be changed for all checked off records. Leaving a field blank will not change the record.




5. Select the correct field for mass update ("status" in this case)
6. Save

CHAPTER 6

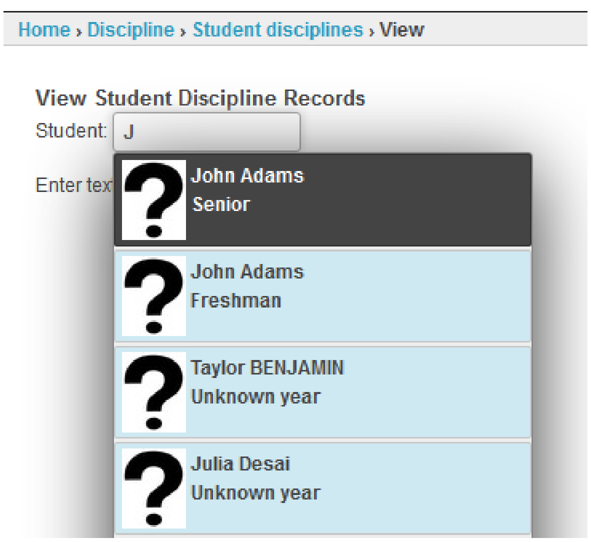
# Discipline

The discipline module tracks a student's discipline information including infractions, actions to be taken, and the teacher who reported the infraction. Similar to the other modules in Django-SIS, discipline reports can be generated and exported into an Excel document.

## 6.1 View Discipline

For fast lookup of a particular student's discipline record:

1. Select **Discipline** from the navigation menu, then **View**.
2. Begin typing in the name of the student in the text box, and Django-SIS will present you with a list of available students as shown below:




3. Once a particular student has been selected, Django-SIS will present all discipline information that has been input for the student:

| Discipline statistics for 2012-2013 |
|---|
| *Action Totals* |
| Detention: 1 |
| Kept phone for a day: 2 |
| *Infraction Totals* |
| Dress Code: 1 |
| Using phone in class: 1 |
| Non Participation in PE: 1 |

The View Discipline screen will display all disciplines that have been recorded for a particular student, and will display the information shown here.

To the left is a yearly tally of discipline statistics, with detailed information about every infraction shown below.

| Date | Teacher | Infraction | Action | Comments |
|---|---|---|---|---|
| Apr 11, 2013 | Teacher, Amy | Non Participation in PE | Detention (1) | Student did not want to dress out into PE clothes. |
| Apr 02, 2013 | Teacher, Amy | Dress Code | Kept phone for a day (1) | Forgot to wear tie. |
| Feb 06, 2013 | Teacher, Amy | Using phone in class | Kept phone for a day (1) | Caught student texting |

## 6.2 Discipline Reports

Displine Reports allows users to pull and filter discipline data by action, infraction, time, and minimum number of incidents.

IMAGE

**By Student Report** produces a list of students who have a record of disciplinary action including details about the incident.

**Aggregate Report** generates an Excel document tallying each disciplinary incident.

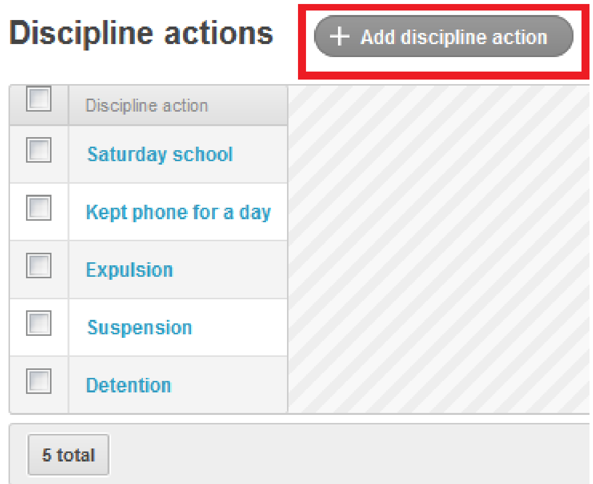
| *Discipline Stats* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Dress Code** | **Using phone** | **Cheating** | **Non Participation** | **Disruptive** | **Detention** | **Suspension** | **Expulsion** | **Kept phone** | **Saturday school** |
| 2 | 1 | 0 | 1 | 0 | 6 | 0 | 0 | 2 | 0 |

## 6.3 Discipline Actions

The link to **Discipline Actions** is located in **Home** under **Discipline**.

Here disciplinary actions available from the dropdown menu may be modified. Clicking **Discipline Actions** presents a list of current discipline actions. To add an action, click **+Add Discipline Action**, enter a new discipline, then Save.

*Schools beginning to use Django-SIS should add all discipline actions that the school currently utilizes.*

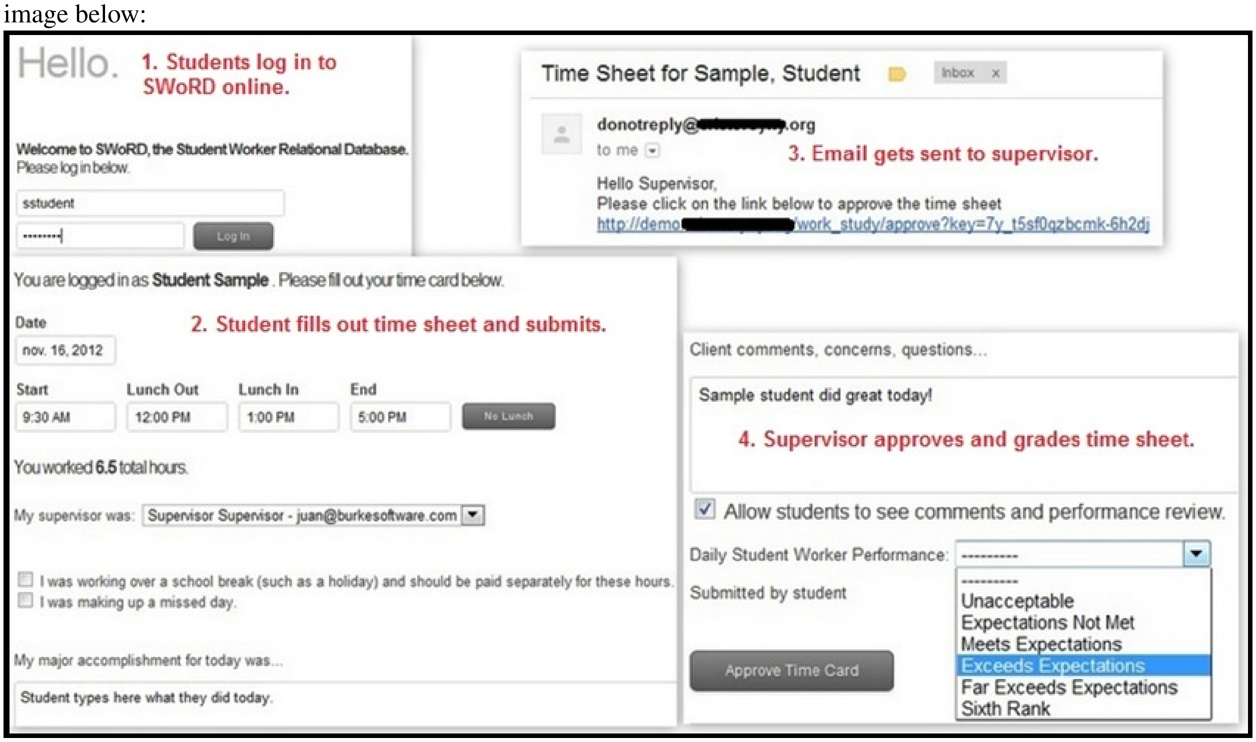

Discipline actions
+ Add discipline action
Discipline action
Saturday school
Kept phone for a day
Expulsion
Suspension
Detention
5 total

# Work Study

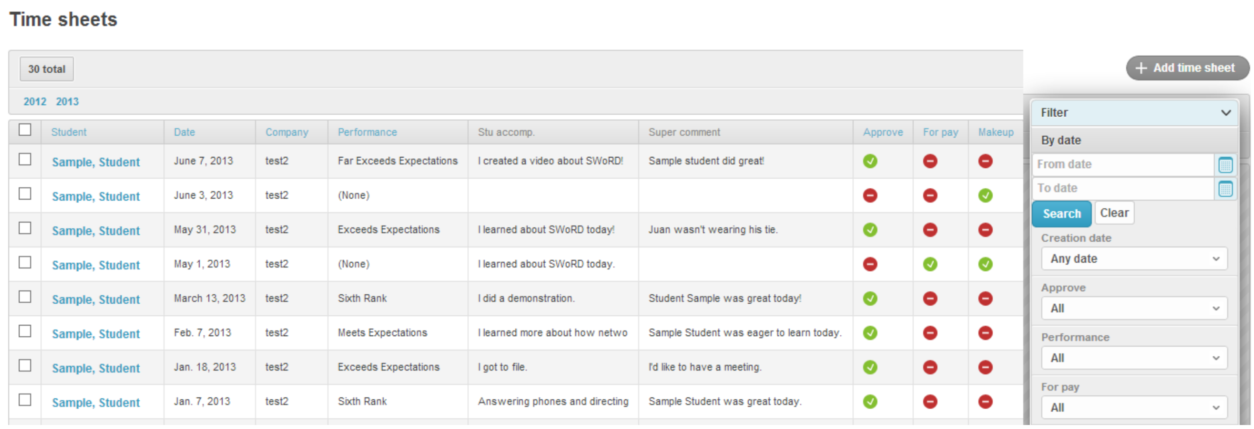
The Django-SIS work study module allows users to keep track of student worker information, including: detailed company information, work assignments, work attendance, directions, work teams, performance reviews, time sheets and more.

## 7.1 Electronic Time Card System

At the heart of Django-SIS's work-study module is the electronic time card system. The electronic time card system allows the school to keep track of a student's working day, what that student did at work, and how their work supervisor felt the student did on a particular day. Additionally, all time cards created by students will be stored neatly into Django-SIS where work-study staff can then filter or create reports accordingly. The basic steps are outlined in the image below:



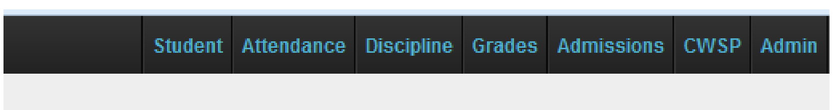

Django-SIS then stores all timecards in the main timecard dash. Users with access to these stored time sheets are able to view each time sheet's information including date, hours, student accomplishments, and supervisor comments.




## 7.2 Adding a Student Worker

To create a student worker, staff members are only required to add a student. All students in the software will immediately be made into student workers.

1. Access the student dash by selecting **Student** at the navigation bar, then **EDIT STUDENTS**



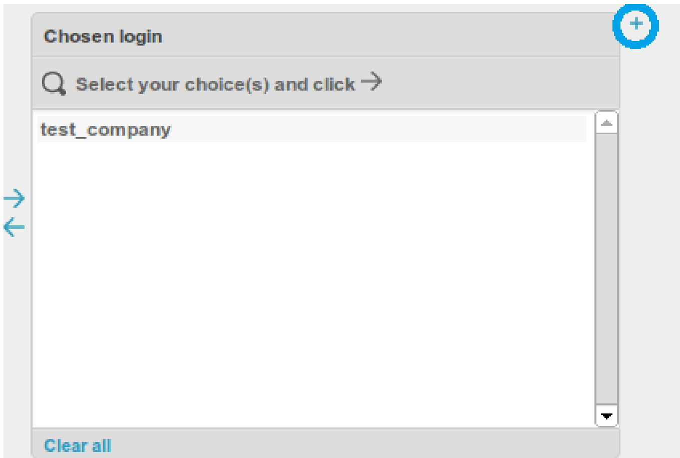

**Navigation Bar**

2. At the list of students screen, select **Add Student** at the top right of your screen.
3. The only **required** information when adding a student is the username. Everything else is optional.
4. Select **Save** at the bottom right of the screen.

You may now access the list of student workers by clicking CWSP at the top menu bar, then clicking on **student workers** under the **EDIT WORK STUDY** dashlet. Your newly created student/worker will now be available to assign work placements, supervisors, etc.

## 7.3 Creating Supervisor Logins

1. Under the CWSP section from the main Django-SIS dash, select **work teams** below the **edit work study** dashlet and click on your desired work team.
2. Select an available login, or click the blue plus located to the right of the box.




**Things to keep in mind**

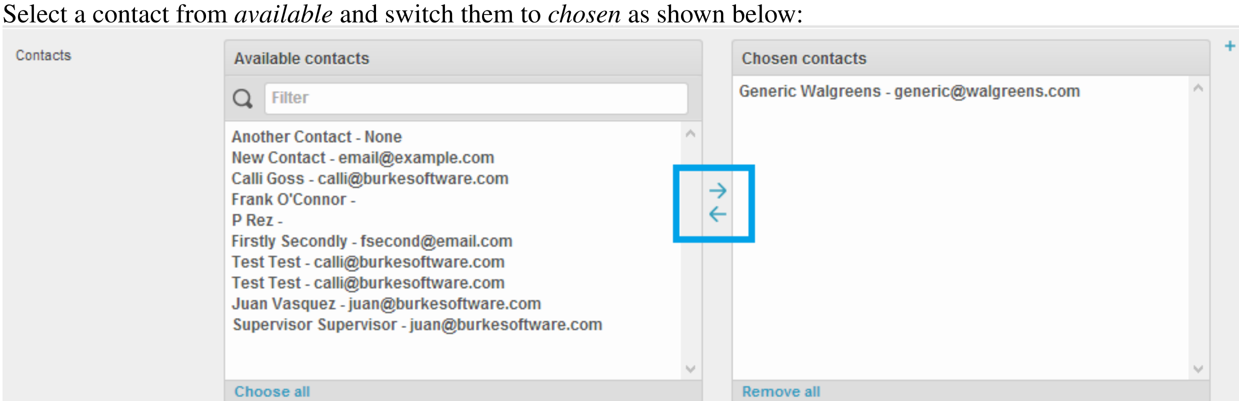
- You need access to create users.
- Supervisors must log in to the base site, not the admin site.
- Do not mark these users (Company) as Faculty or Student users. Doing so will produce unexpected results.
- One work team may have an unlimited number of supervisor logins.
- Supervisor login is not related to the supervisor contact in anyway.

## 7.4 Creating Contact Supervisors

Contact supervisors are individuals at companies that supervise a school's student worker. Accordingly, these contact supervisors are those people who will be receiving emails to evaluate/appraise a student worker's time card that has been submitted to them. To set them up, follow the directions below:

1. Under the **Edit Work Study** dashlet from the CWSP Django-SIS dash, select Add+ by **Contact Supervisors**
2. Fill out the available data fields.
3. Save

Once the contact has been created, that contact may be assigned to any number of work teams. To assign your newly created supervisor to a work team, click on **Work teams** under *Company Data* and scroll down to the **Contacts** box. Select a contact from *available* and switch them to *chosen* as shown below:




**Don't forget to click SAVE after assigning a Contact**

## 7.5 Work Teams

Work teams are a single student, or group of students that are assigned jobs at a company. Work teams are especially helpful for when a school has multiple students working at the same company, but have different types of jobs at that company. For example,

Student A and Student B both work at Sample Company, but in different departments.

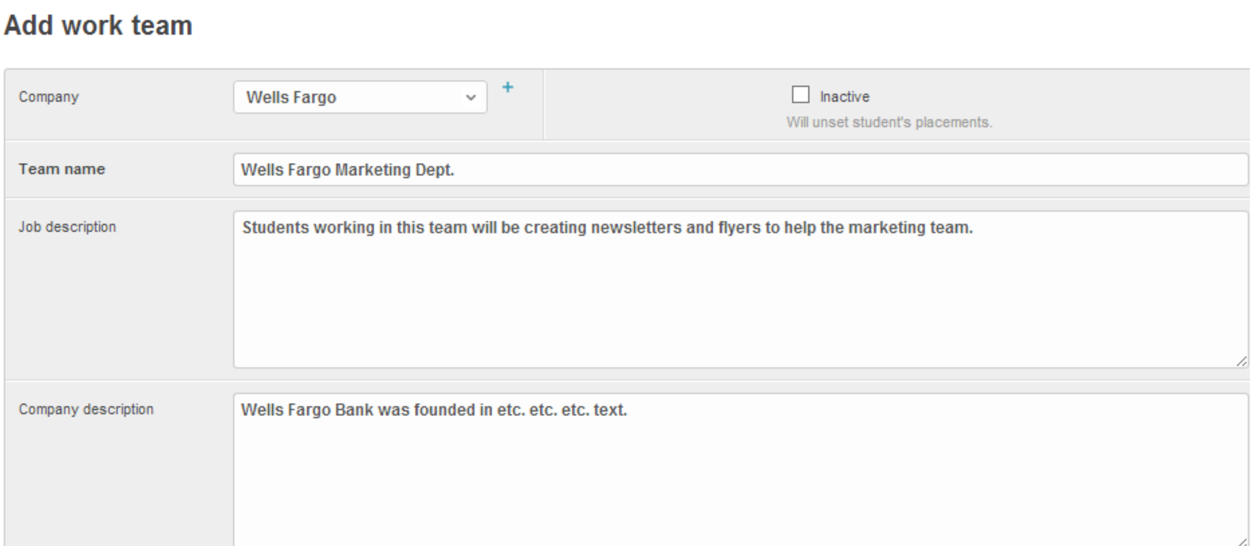
Work teams allow schools to create unique work teams that are still associated with the same company:

Student A - Sample Company Front Office Student B - Sample Company Marketing Dept.

**Creating Work Teams**

1. Select Add+ by **Work Teams** located under the *Edit Work Study* dashlet.
2. Fill out the information available. A team name is *required*.
3. Save

In the example image below, a work team, *Wells Fargo Marketing Dept* is created and is associated with the company, *Wells Fargo*

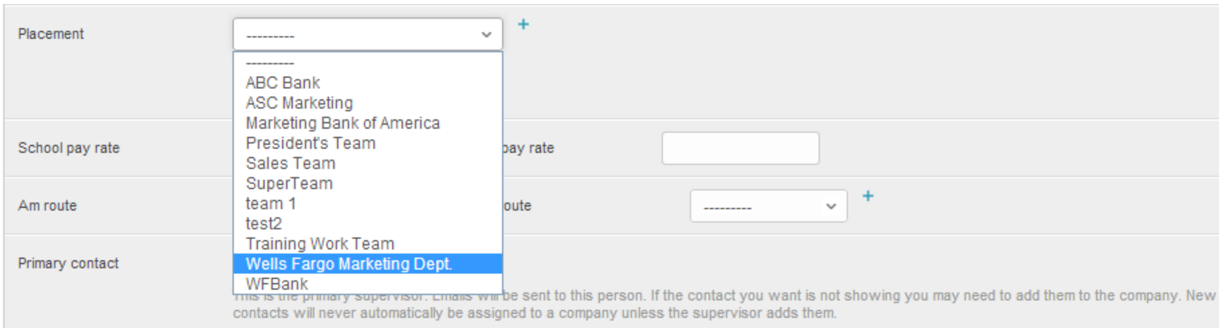



## 7.6 Assigning Work Placements

Once all the work teams have been created, you can now assign student/s to a work team. To do so, follow the directions below:

**For a single student**

1. Click on student worker's name
2. Assign the student a placement from the dropdown box. *Note*: placements refer to workteams (see above)



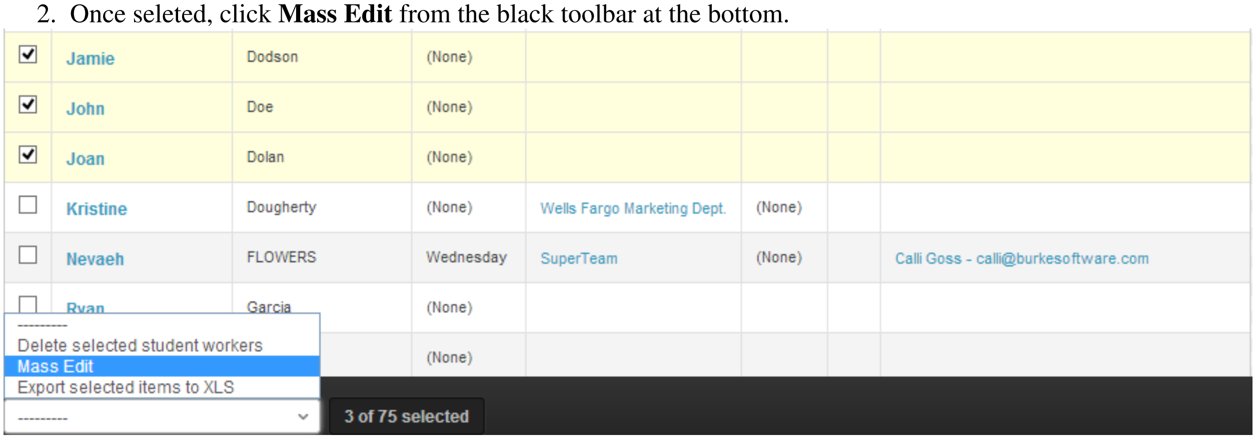

3. Save

Once saved, you will see the changes reflected immediately in the student worker dash:

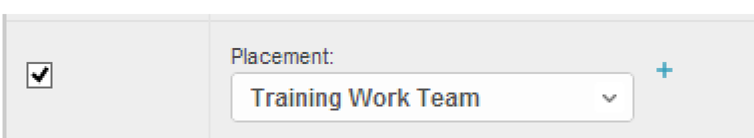

### Multiple Students

Using the Mass Edit function, work study staff can assign multiple students to the same work team quickly, instead of assigning by individual student. To do so:

1. Go to the **Student Worker** dashboard, and select your students to group assign placements.
2. Once seleted, click **Mass Edit** from the black toolbar at the bottom.

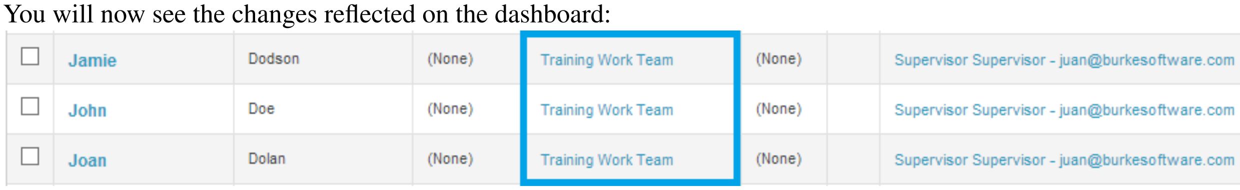



3. At the mass edit screen, check the box by placement and assign a placement.



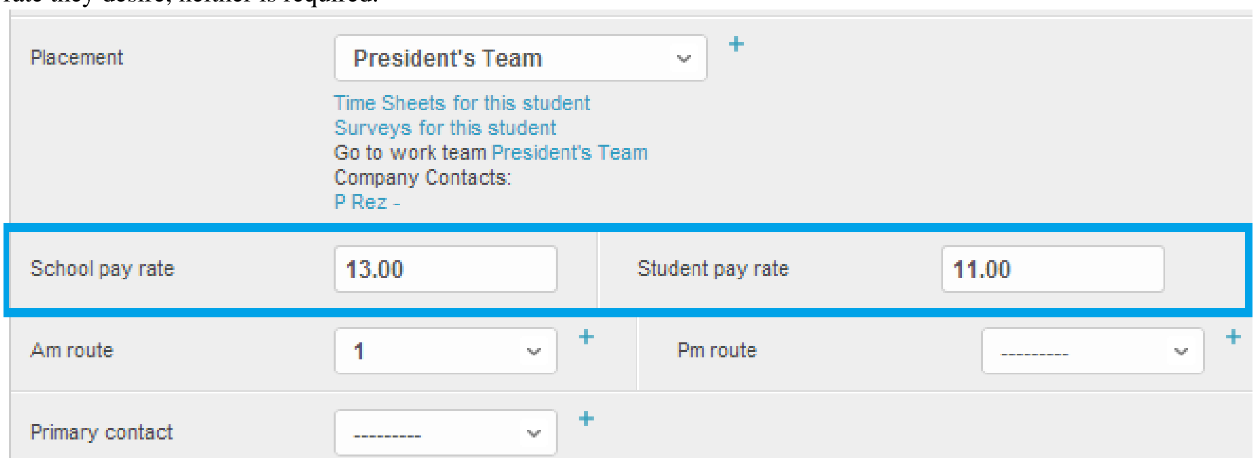

4. Save

You will now see the changes reflected on the dashboard:



## 7.7 Student Pay Rates

School staff are able to set a pay rate that an individual student and a company gets per hour. The two options shown below exist for instances in which a school takes an accounting fee cut of the paycheck. Individuals can set either pay rate they desire, neither is required.

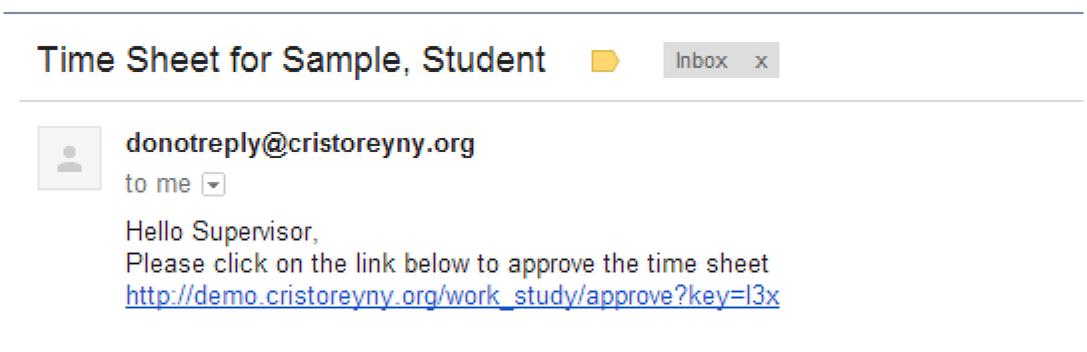



**Note** Schools have the option of setting a default pay rate in Django-SIS's configurations. This price will by default appear on time sheets and student worker pages, including being a default when new students are created.

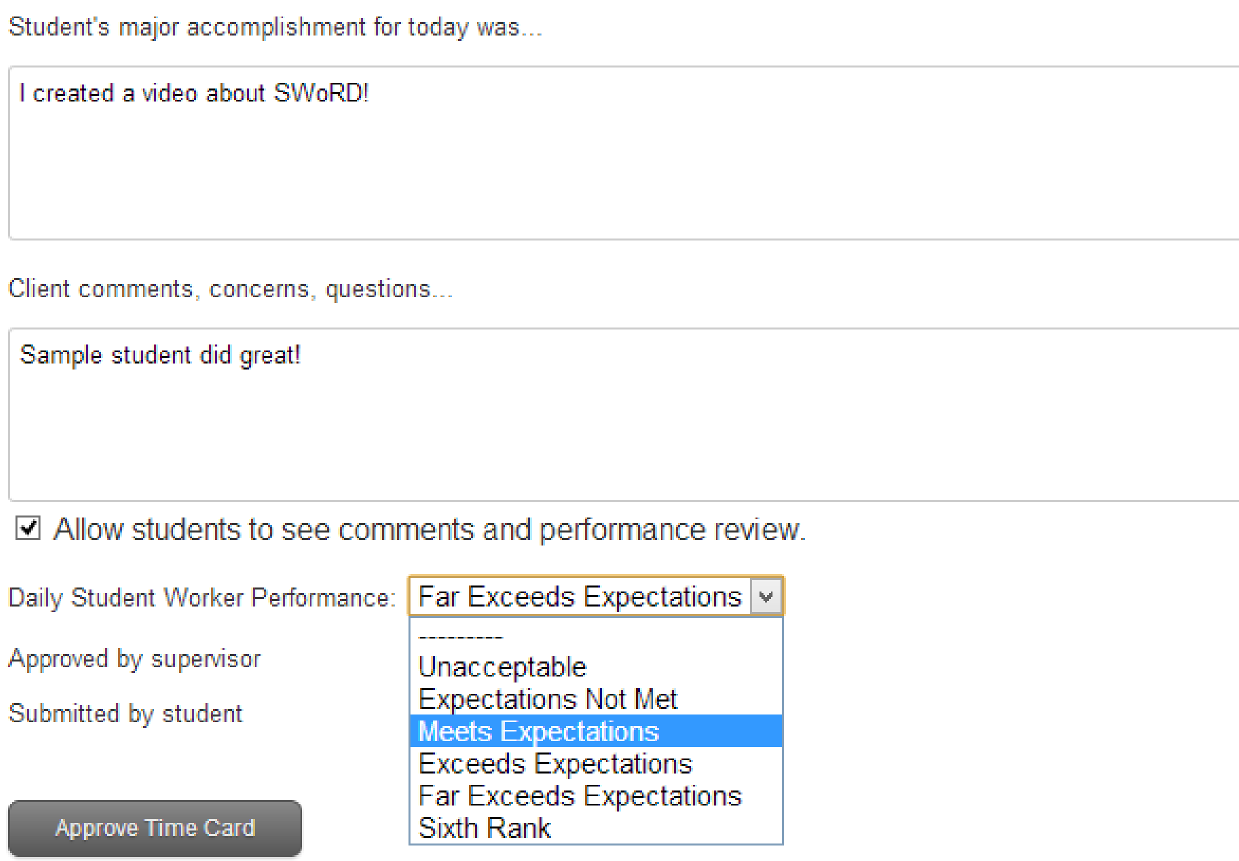
In instances where certain students get paid differently than others, you can edit the student/s by clicking on their individual student worker page as shown above, or mass editing the change from the student worker dashboard.

## 7.8 Supervisor View/Timecard Approval

Once the student submits a timecard for approval, an email will automatically be sent to the student's primary supervisor asking for approval, as shown below:



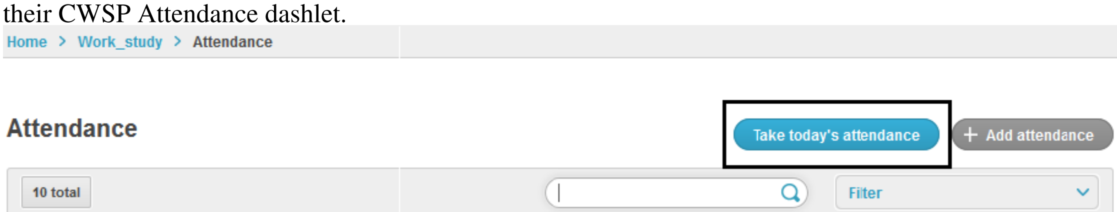

The email will instruct the supervisor to click on the special link provided. Once selected, the supervisor will then be sent to the time sheet the student submitted that displays what the student did, time in, time out, etc.



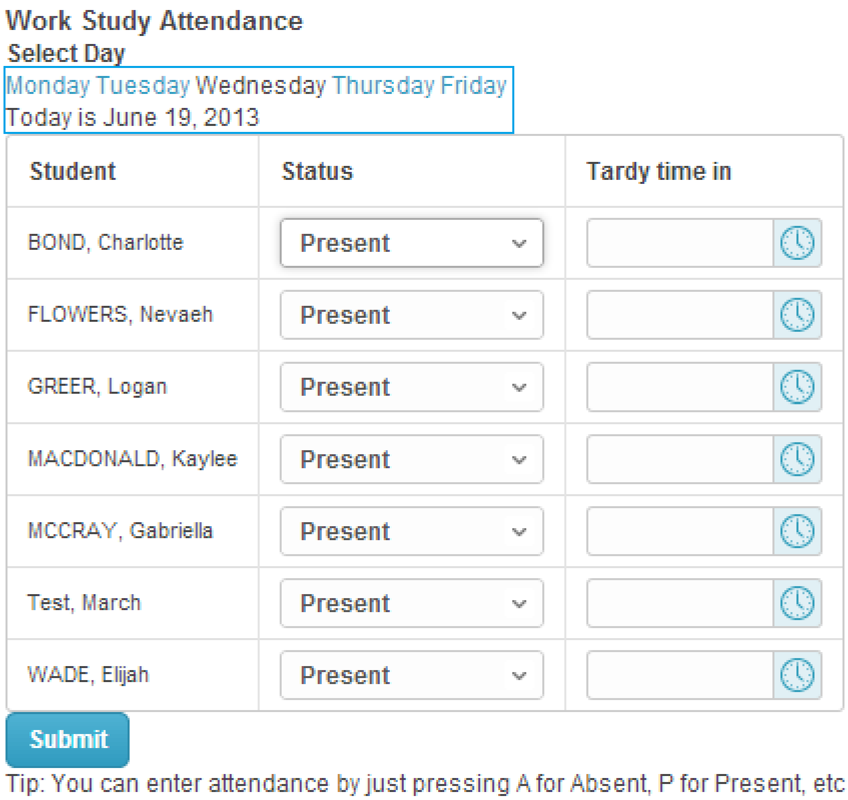

From here, the supervisor may write questions/comments in the provided text box, as well as provide an evaluation from a drop down box- these options may be customized to fit a school's need. When the supervisor approves the time card, work study staff will see it marked as approved in the time sheet dashboard.

## 7.9 Work Study Attendance

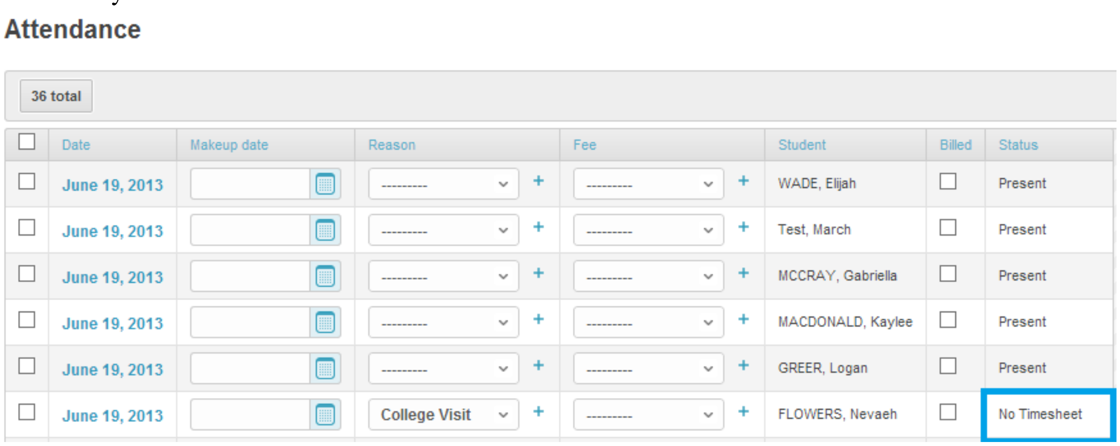
Work Study staff members are able to track their own attendance. By selecting "Take today's attendance" button from their CWSP Attendance dashlet.




Django-SIS will then display a list of students whose working day is that particular day, as shown:




Once submitted, Django-SIS will then display both Present students and Absent students in the dashboard, in addition to Tardy and Absent/Half Day as they are marked by homeroom teachers. This allows work study staff to determine whether or not a student should submit a time sheet.



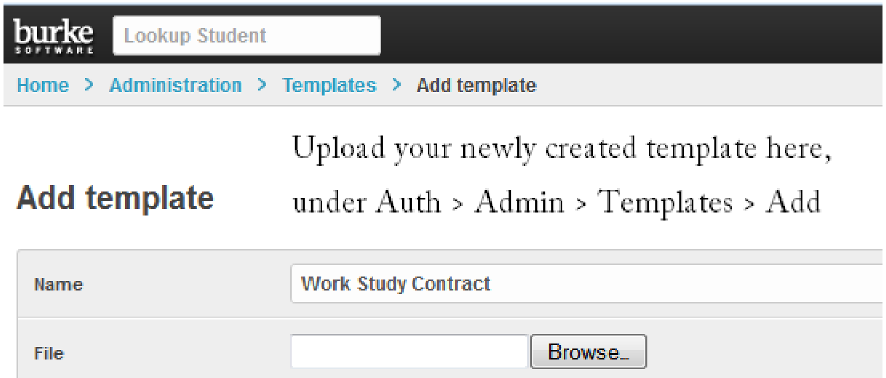

## 7.10 Electronic Contracts

Django-SIS supports the ability to store and sign fully electronic work study contracts between the school and the student's work placement. Contracts can be filled out by the client from a web interface, or manually added to a company under the **Companies** selection from the CWSP header.

**Note:** Contracts are linked to *companies*, not work teams.

1. For fully electronic contracts, start by editing or creating a template, and be make certain that it's named "Work

Study Contract". If your school has purchased Django-SIS support, feel free to email for assistance in this regard, otherwise refer to the "Report Writing and Creating Templates" section above to do so indepedently.




2. Next, you will need to generate special web links for **each** company you want to get a contract from. It is recommended that you try this first with a fake companay to see how it works.
3. To get to this link, enter <site url>/work_study/company_contract/<company id #> Where site url is replaced with the URL for your Django-SIS installation. Company ID # can be downloaded with the export to XLS tool. Go to **companies**, check off the companies you want to get IDs for, click **Export to XLS** and make sure the ID column option is checked.

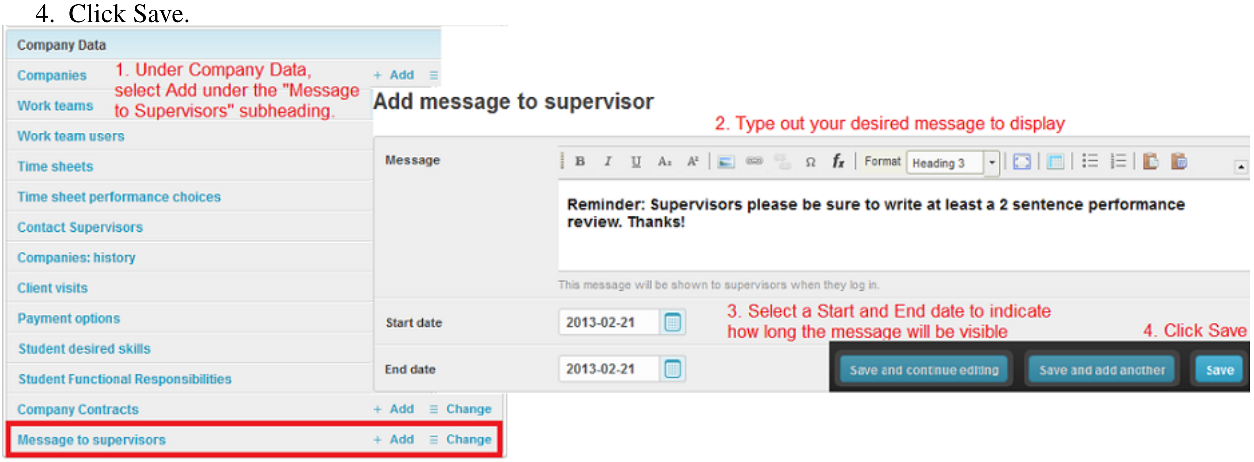
A finished url might look something like: sis.YOURSCHOOL.org/work_study/company_contract/123

You may send this link to your client and wait for the results to come in. Because Django-SIS stores contracts, you may review it once it has been submitted under Work_study > Company Contracts.

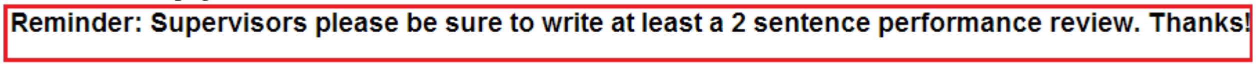
## 7.11 Message to Supervisors

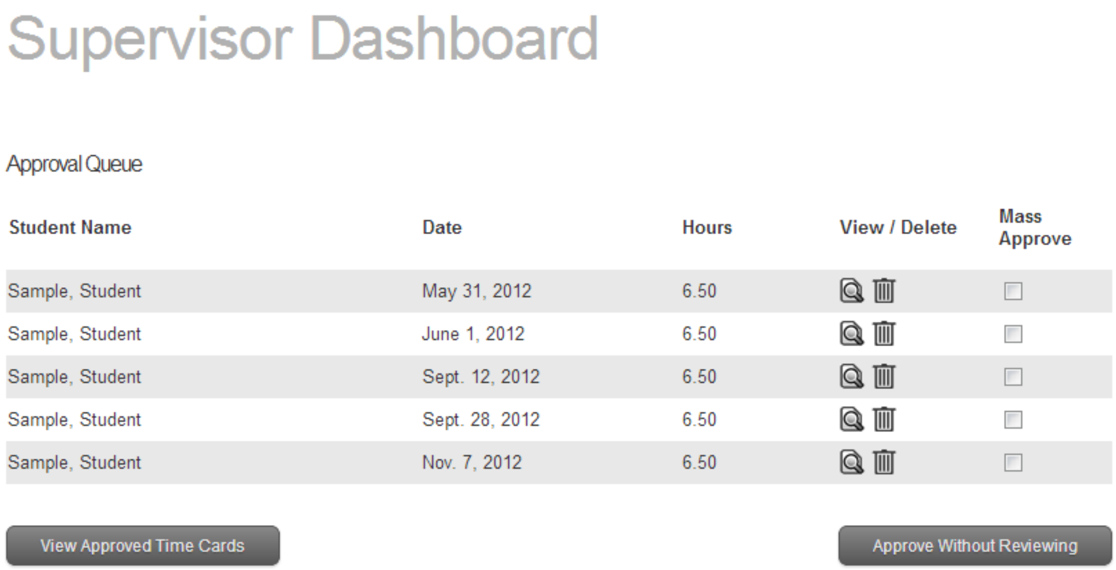
Django-SIS allows school officials to send out messages/reminders for all supervisors to see when they log in to approve a student's time card. The steps are as follows, with an attached picture depicting the steps and outcome:

1. Under **Edit Work Study** dashlet, select ADD under the *Message to Supervisors* subheading.
2. Type out your desired message to display to all supervisors.
3. Select a Start and End date to indicate how long the message will be visible.
4. Click Save.



After the previous steps above have been completed, Django-SIS will then display your message to supervisors on their dashboard page, as shown below:



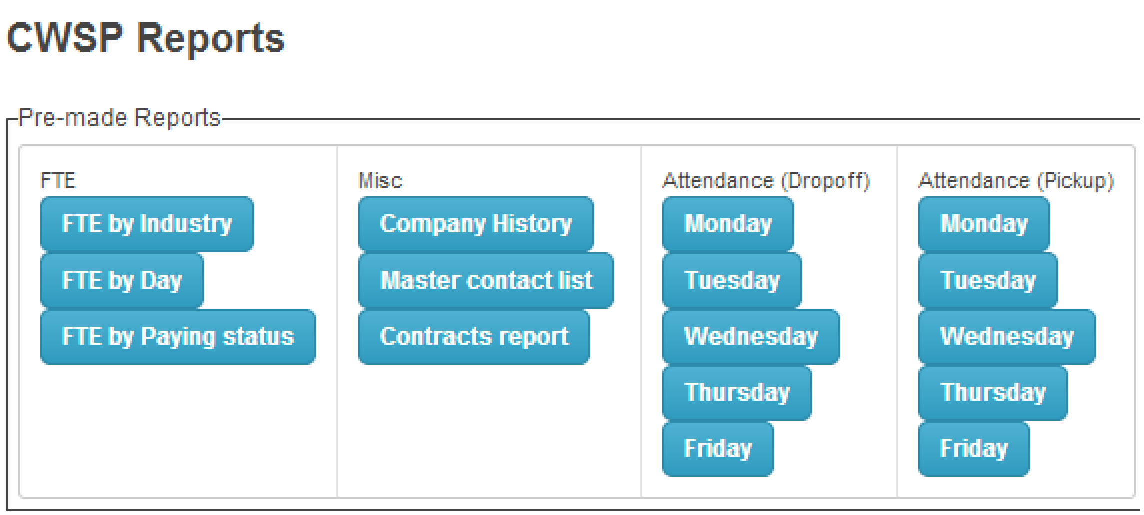




## 7.12 CWSP Reports

Similar to other modules, the Work Study module in Django-SIS comes packaged with a number of pre-built work study reports. In addition to the pre made reports, there is a section available for template based reports which will generate reports that a user has created. A description of the three major types of reports, along with examples will be shown below:

**Pre-made Reports**

The pre-made reports are one click reports that cover: FTE, MISC, Atendance dropoff, Attendance Pickup.




*FTE reports:* (Full-time equivalent): by industry, day, and paying status will generate an overview and a per student look covering the aforementioned filters.

**MISC**

*Company History:* Will generate an Excel document detailing all student placements at a company by date.

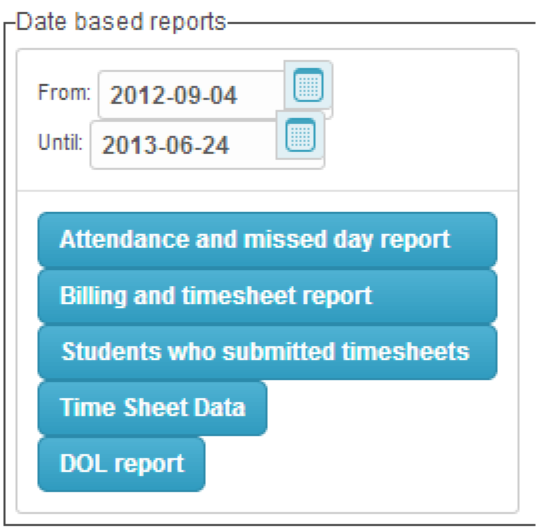
*Master contact list:* Produces an Excel document showing all student contacts- their work contacts number/email and each parent email/phone number.

*Contracts report:* Excel document showing each company, whether or not there is a contract, and when the date for the last contract was recorded.

*Attendance Dropoff and Pickup:* Shows each student worker attendance that is working on the designated day you click, along with transportation information (subway line, stop location) and associated company.

**Date based reports**

These reports require the user to set up a date range. Once the date range is set, the user may click on a report in which Django-SIS will tailor to the dates accordingly.




*Attendance and missed day report:* Multi tabbed Excel document that displays students who missed their work day, and if/when that work day will be made up along with comments and totals. Additionally, a separate tab will display all student worker time sheets that were submitted during the date range.

*Billing and timesheet report:* Produces a billing report for the date range, including hours worked, and amount to be billed (multiplies hours worked by school pay rate to arrive at a total) to each company. This report breaks the billing and timesheet reports down by student.

*Students who submitted timesheets:* lists name of all students, and how many time sheets they submitted (including dates) during the date range set.

*Time Sheet Data:* Detailed look at all submitted time sheets during the specified date range.

*DOL Report:* Displays a school work study employee's visits to client companies for the date range.

**Template Based Reports**

These reports allow users to generate reports that they have created themselves, although Django-SIS does come packaged with some already made generic templates, including a travel maps/directions to help work study students get to their work site, as shown below:

## Travel Directions

STUDENT

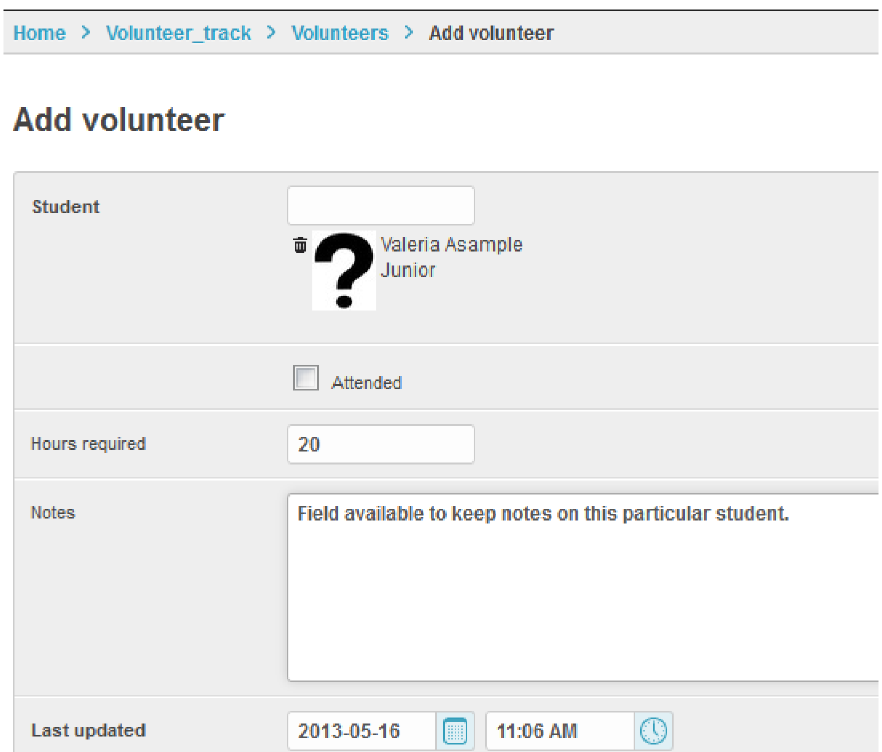
| | |
|---|---|
| Company: Jones Day LLP<br>Address: 222 E 41st Street<br>New York, NY 10017<br>Train: 6<br>Stop location: 42nd to exit<br>Company Contact:<br>Sample Company Contact Name<br>Sample phone number | Directions to:<br>Take the 6 Train at the 103rd Street station headed downtown until you reach the 42nd Street Station (Grand Central). Walk south on Park Avenue until you reach 41st Street, turn left and walk down past 3rd Avenue and look for address 222.<br>Directions to pickup:<br>Walk back to the 42nd Street Station. Meet chaperone at entrance to in between 456 trains and "S" shuttle |

# Volunteer Tracking

Some schools require students to complete a certain number of volunteer hours every school year. Accordingly, Django-SIS allows school staff to keep track of a student's volunteer hours, sites, and site supervisors. Tracking volunteers works similar to other modules in terms of adding and storing data.

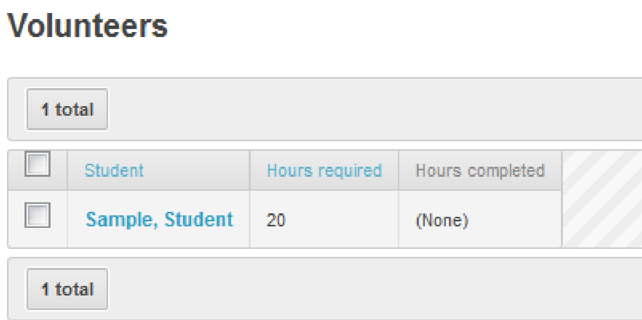
## 8.1 Adding a Volunteer

Locate the **Volunteer_track** module of Django-SIS from your main dashboard screen. Once found, select **add** by the **volunteers** option. The following page will be displayed:




Under **student** begin typing in the name of a student you will be adding as a volunteer. A list of students will then show in a drop down box. Once your selection has been made, the remaining fields are optional- hours required, notes, sites. Select **Save**.

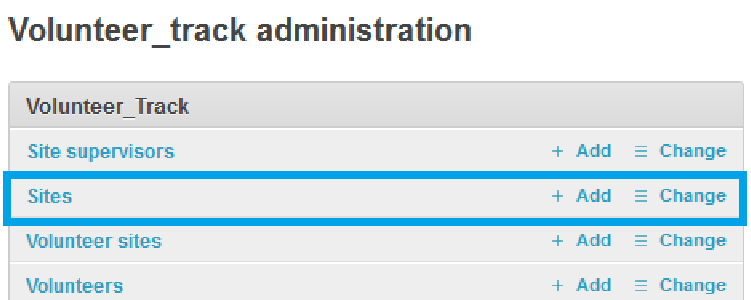
Volunteers will be stored under the **Volunteers** heading along with their progress in number of volunteer hours completed.



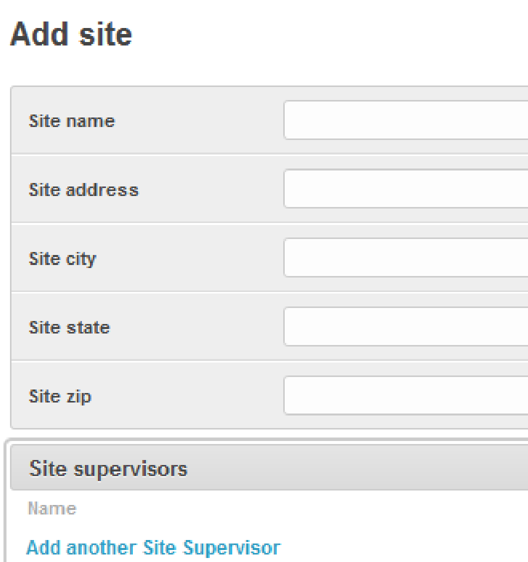

## 8.2 Add a Site for Volunteers

**Sites** refer to the physical location of where students will be volunteering. In the volunteer track module, the **volunteer sites** option is for school staff to create a new volunteer session for a student without the student submitting.

From the **Volunteer Tracking** header select +Add by the Sites option.

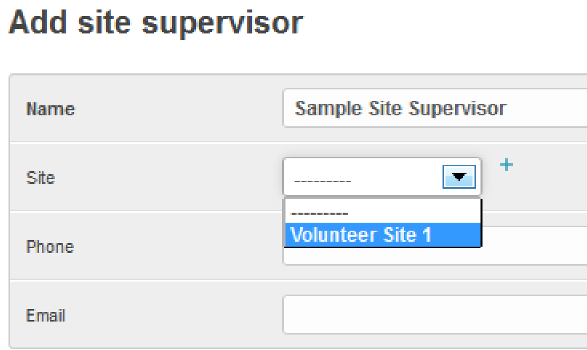



Next, you'll see the image below directing you to fill out basic site information. Save your changes.



## 8.3 Assign a Site Supervisor

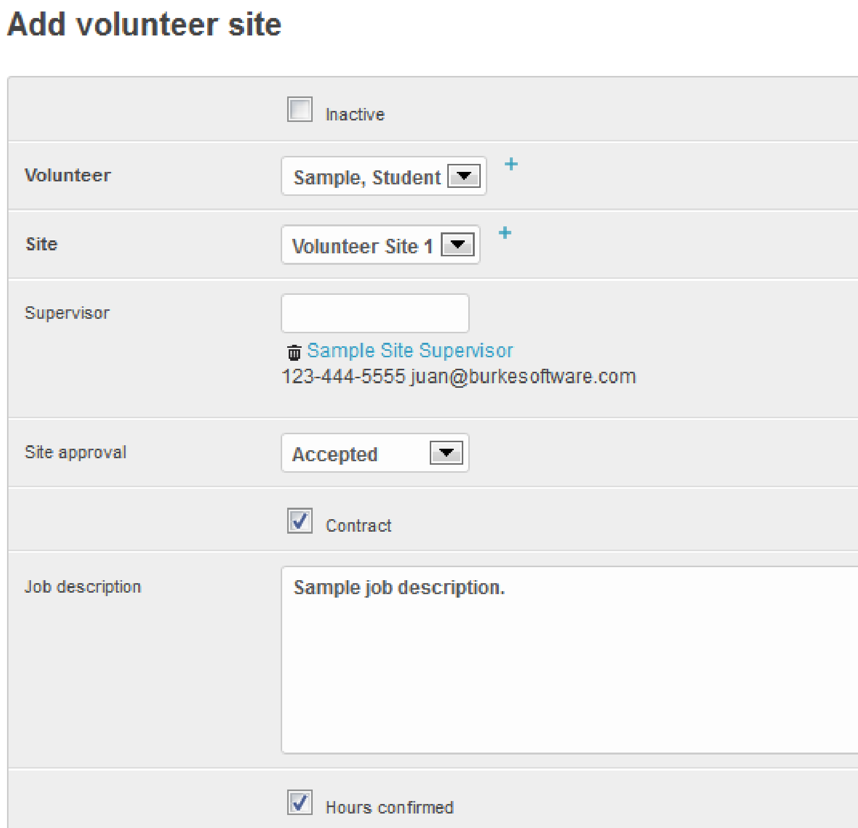
At the familiar **Volunteer Tracking** module, select +Add next the **Site supervisors** option.



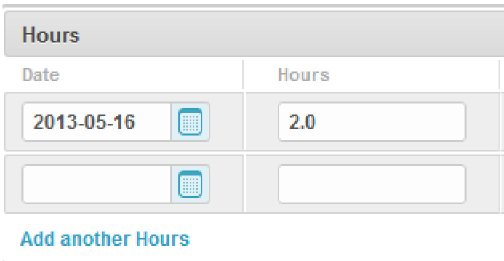

From this screen, add your information in about the supervisor. Note: only the NAME field is required, although ideally you could set the **site** of where this person is in charge of at this screen as well.

## 8.4 Add Volunteer Site

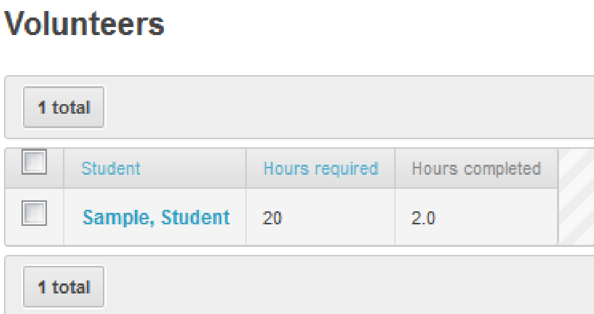
Selct +Add by the **Add Volunter Site** on the Volunteer Tracking module. Clicking add will lead to the following screen:




Here, you may enter the appropiate information in to register hours for a particular student.




Once the hours have been registered, you will see this reflected in the volunteer dash.



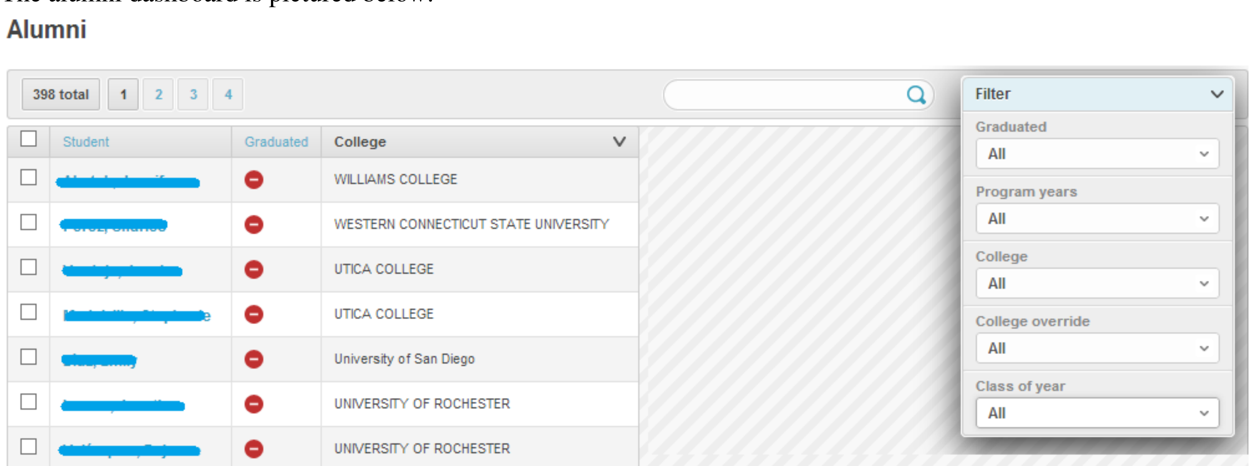
# Alumni

The alumni module allows schools using Django-SIS to keep track of students who have graduated or left their respective schools. When schools increment school years in preparation for the new year, all classes at that time will move up one year (Freshmen become Sophomores, etc.). Seniors will then graduate and become alumni in the system. Optionally, Django-SIS can sync with National Student Clearinghouse to help track online.

## 9.1 Alumni Dashboard

You can access the alumni module by clicking on Alumni > Alumni Administration at the top menu bar in Django-SIS. From here, select **Alumni** to get to the dashboard.

The alumni dashboard is pictured below:




This dashboard allows users to gather and export a list of alumni as well as utilize filters to quickly pull pertinent data.

**Alumni Filters**

In sum, users have 5 filters available to utilize that Django-SIS will sort accordingly:

*Graduated*

Whether or not the student graduated from your school

*Program Years*

What type of program that student went on to advance to (2 year college, 4 year, etc.) after graduating,

*College*

Name of the college the student is attending

*College override*

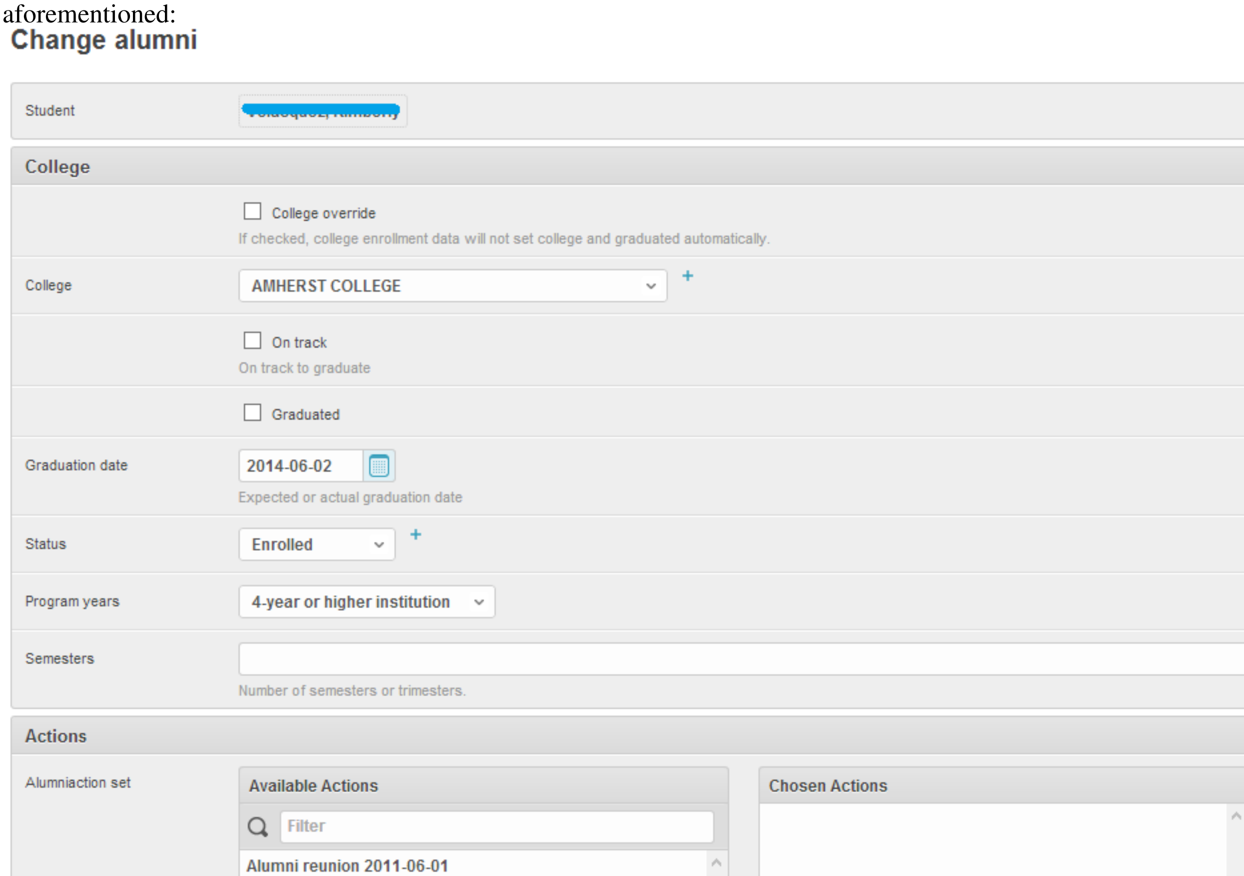
Yes/No field where checked indicates college enrollment data will not set college and graduated automatically. Most likely the majority of alumni you have will not have this checked.

*Class of year*

What year the alumni graduated from your school.

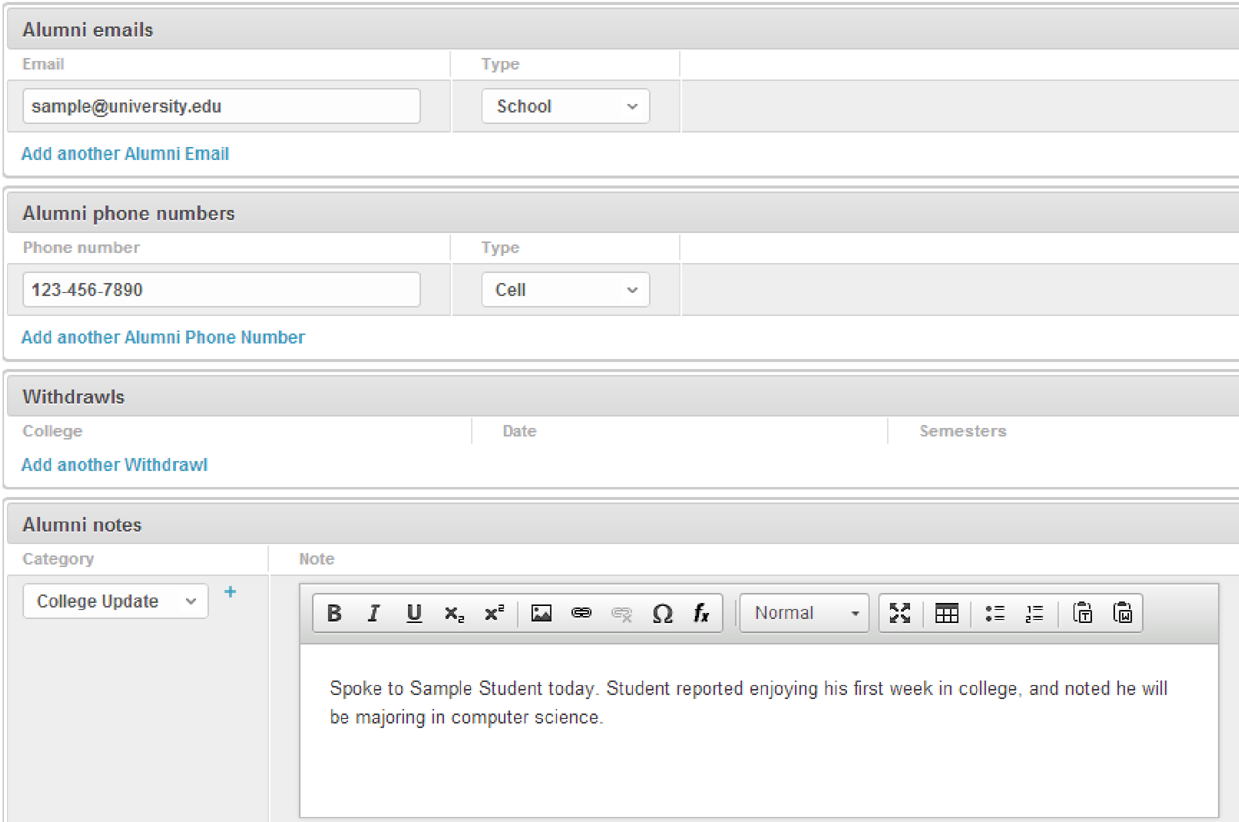
## 9.2 Storing Alumni Data

Clicking on an alumnus from the student dashboard opens up a page that allows users to enter a multitude of data. Towards the the top is basic data, including: college, status, graduation date, type of college, and alumni action. If you make any edits, don't forget to click **Save** at the bottom right of your screen. The image below details the aforementioned:




**Additional Data**

In addition to the basic data outlined above, the Alumni module allows school staff to store any notes or interactions with that particular student.


Alumni emails
Email
Type
sample@university.edu
School
Add another Alumni Email
Alumni phone numbers
Phone number
Type
123-456-7890
Cell
Add another Alumni Phone Number
Withdrawls
College
Date
Semesters
Add another Withdrawl
Alumni notes
Category
Note
College Update
Normal
Spoke to Sample Student today. Student reported enjoying his first week in college, and noted he will be majoring in computer science.

CHAPTER 10

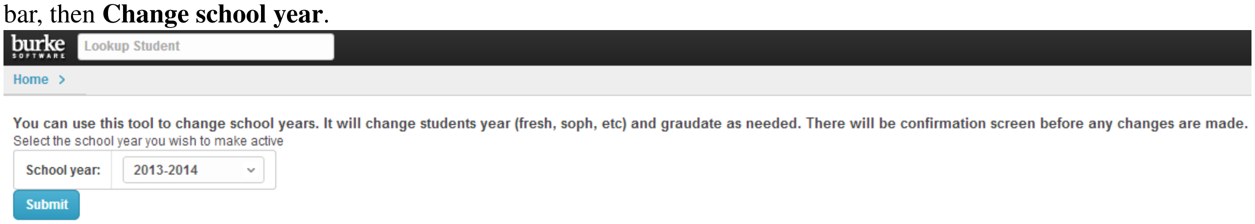
# Preparing for a New Year

This section will serve as a checklist for users as they get ready for a new school year.

1. **Change School Year**

*Note:* This will change students year- freshmen will be come sophomores, seniors will graduate, etc.

How to: User must have the necessary permissions (typically a registrar). Select **Admin** from the top-right selection bar, then **Change school year**.

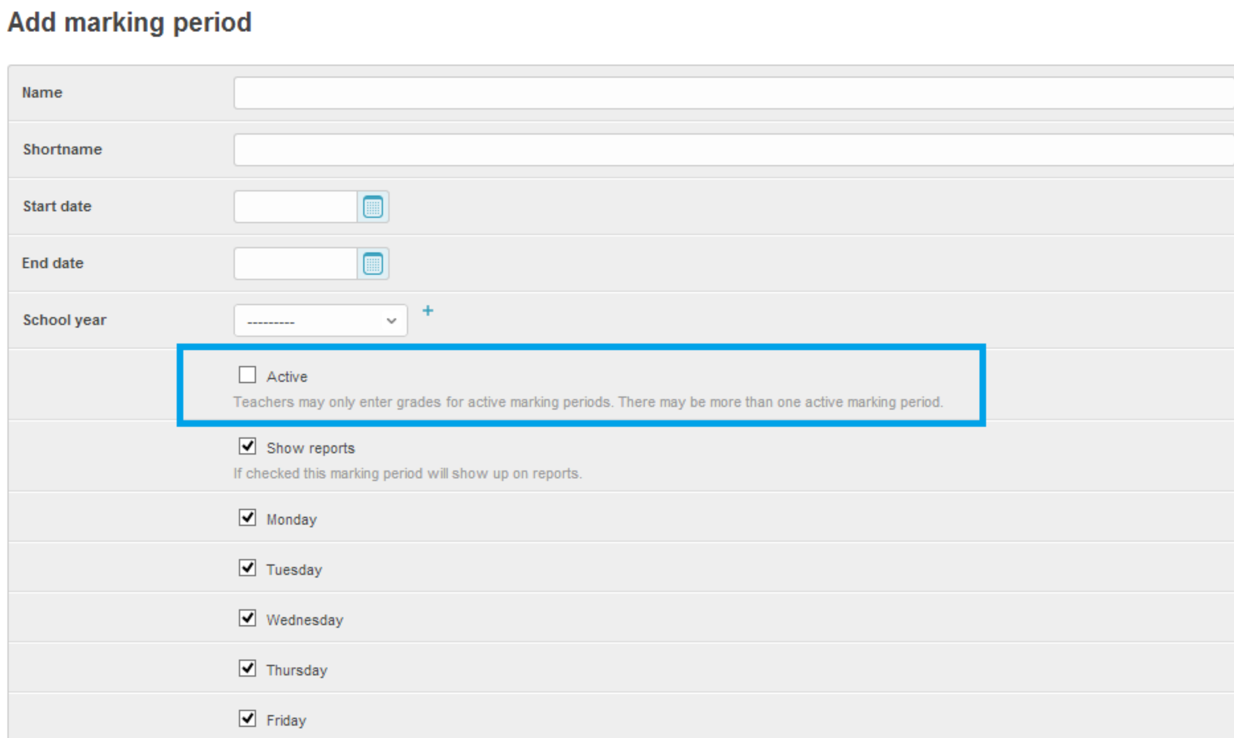



Select the appropiate year from the drop down and submit. A confirmation screen appears and the change will go through once **YES** has been selected.

2. **Create New Marking Periods/Semesters**

In order for grades/attendances/disciplines etc. to be logged appropiately, it is vital that the correct marking periods are set up, and that you set the correct current marking period.

How to: Under the Courses and Grades tab, select **Marking periods**. If you notice that marking periods for the next year have not been created, select +Add. You will be taken to the following screen:




Fill out the appropiate information and set the correct dates. Check the **Active** box if that marking period will be the first marking period for the new year (i.e. August - November). **Save**

3. **Create students from Applicants**

You may refer to **'Creating Students from Applicants'_**. One thing worth repeating: the only applicants who will be converted into students are those applicants who have the green check under the **ready to export** column header.

4. **Create Courses for the New Year**

How to: click here to read over the how-to section on this topic **'Creating Courses'_**.

*Note:* ensure that you select the correct marking period when creating new courses.

5. **Enroll Students in New Courses**

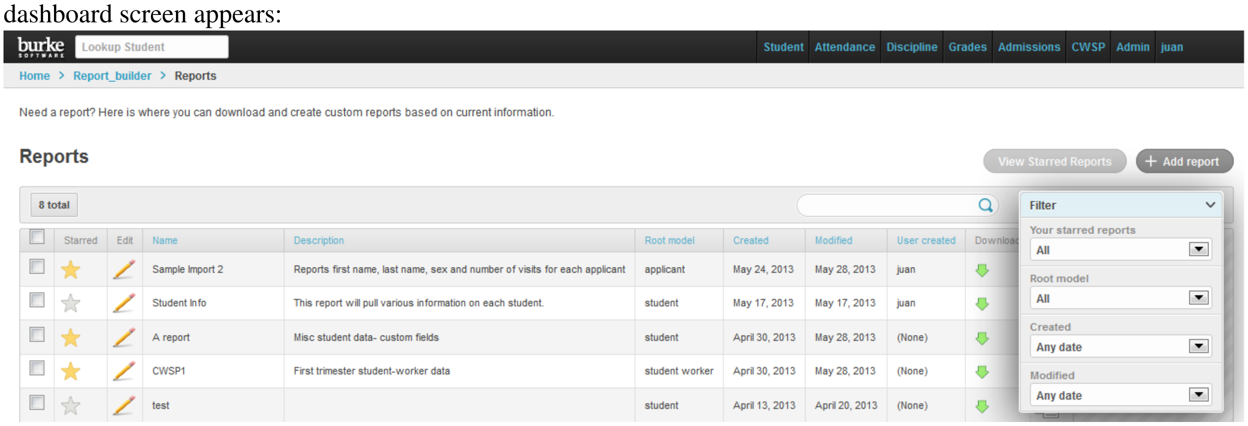
How to: refer to **'Enrolling Students'_**.

# Custom Report Builder

Packaged with every instance of Django-SIS is the custom report builder tool. This tool allows users with permission to easily create custom reports utilizing a drag and drop method. This section will cover how to utilize this tool.

## 11.1 Report Builder Dashboard

Access the admin report builder site (sampleurl/admin/report_builder) and click **reports**. The following report builder dashboard screen appears:

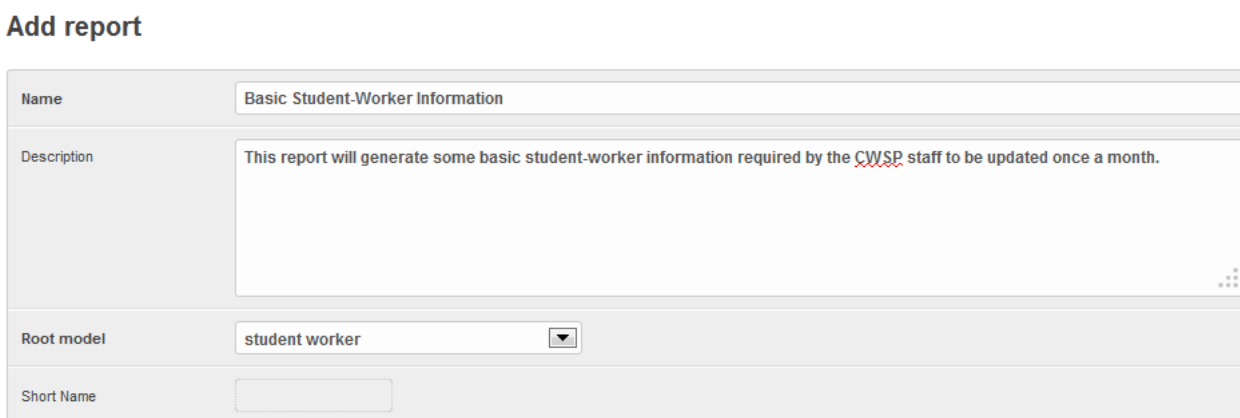



This dashboard will allow the user to view any reports that have previously been created. Additionally, users will have the option of utilizing the available filter to quickly access, sort, and view previous reports by status, date, and root model.

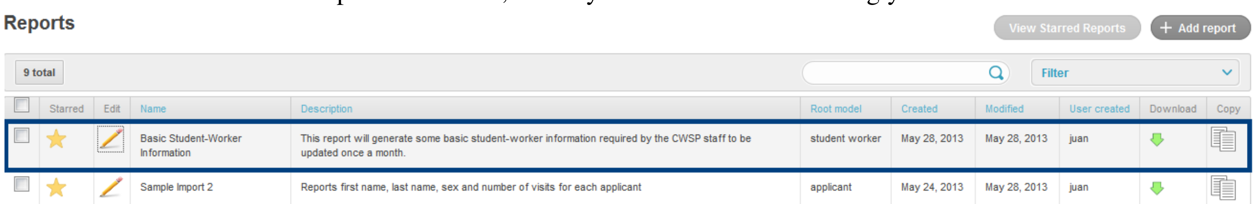
**Starred Reports** are utilized to mark important reports, or reports that will be frequently generated. Users may quickly sort the dash to view only starred reports by selecting the **View Starred Reports** button located towards the top of the dashboard.

## 11.2 Creating a Report

From the report builder main screen described above, select **Add Report** located at the top right-hand corner of the dash. The *Add Report* screen displays- **name** and **root model** (students, applicants, workers, etc.) are required fields.

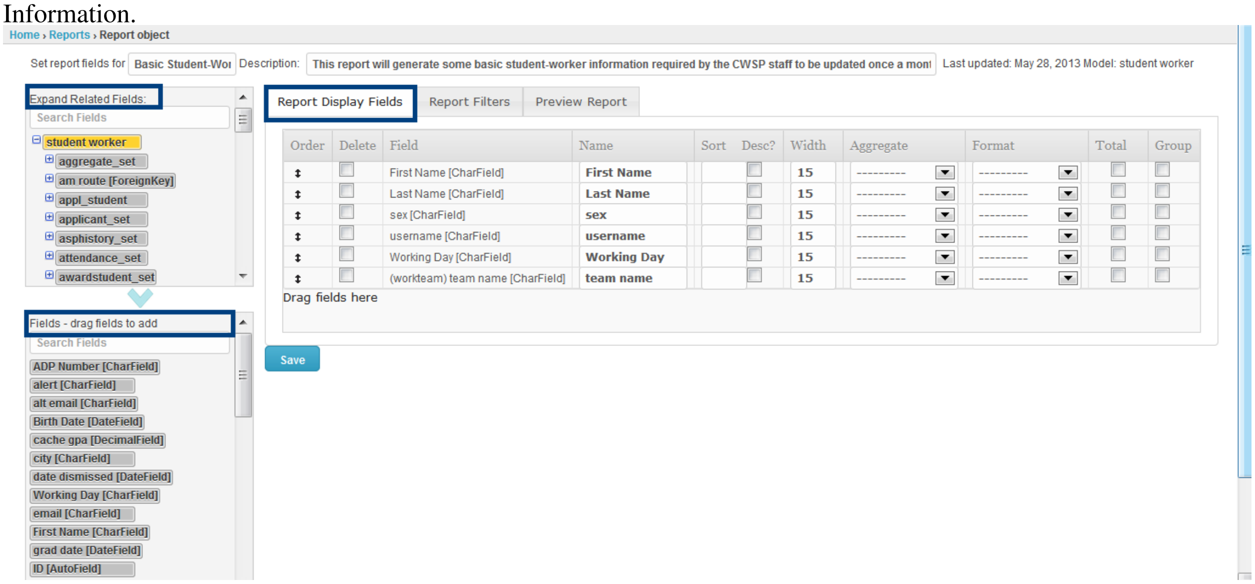



It may also be helpful to include an extended description as shown above to provide other users with a more clear direction of how the report is used. Once the information has been entered, select **Save** Your newly created report will now show as the most recent report in the dash, where you can then edit accordingly:




## 11.3 Editing a Report

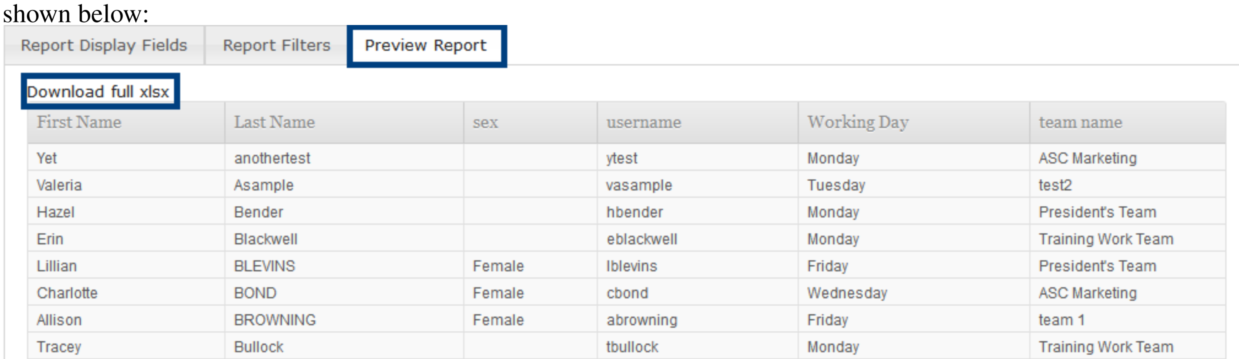
All created reports have the option of being edited. Using the newly created report from above, to begin editing, select the pencil icon located under the **Edit** column by the respective report, in this instance: Basic Student-Worker Information.




With the **Report Display Fields** tab selected at the top, click and drag the fields from the list of available options located at the bottom-left side of the screen into the empty area located directly to the right while the appropiate tab is still selected. **Save** your selection.

*Note:* The **Expand Related Fields** field located in the box above the current fields list allows users to access expanded fields/information. Selecting one option will generate the expanded fields in the box below where you can then drag and drop into the space available accordingly.

After dragging your specified fields and saving, users then have the option to preview the report by selecting the **Preview Report** tab. This will generate a preview, where users can then export into an Excel/Libre-Calc document, as shown below:



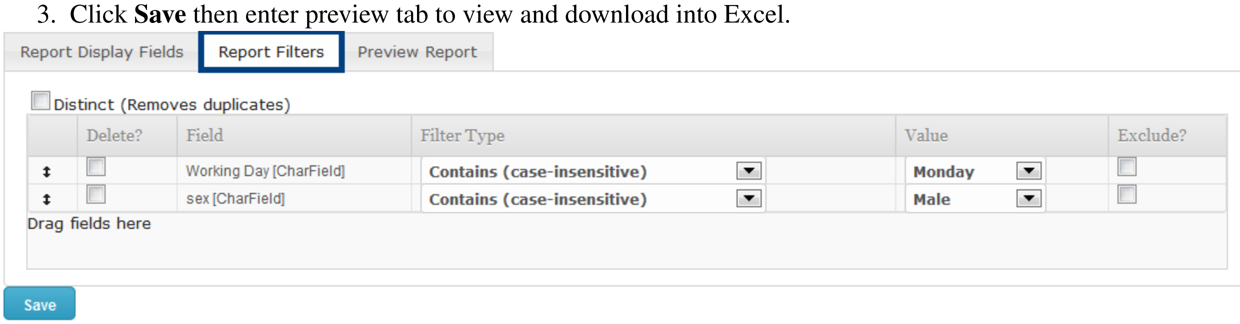

| First Name | Last Name | sex | username | Working Day | team name |
|---|---|---|---|---|---|
| Yet | anothertest | | ytest | Monday | ASC Marketing |
| Valeria | Asample | | vasample | Tuesday | test2 |
| Hazel | Bender | | hbender | Monday | President's Team |
| Erin | Blackwell | | eblackwell | Monday | Training Work Team |
| Lillian | BLEVINS | Female | lblevins | Friday | President's Team |
| Charlotte | BOND | Female | cbond | Wednesday | ASC Marketing |
| Allison | BROWNING | Female | abrowning | Friday | team 1 |
| Tracey | Bullock | | tbullock | Monday | Training Work Team |

## 11.4 Using the Report Filters Tab

The **Report Filters** option is designed to give users the ability to further refine their data. Refining information functions similiar to editing your report, simply select the **Report Filters** tab and use the available fields list on the bottom left to drag and drop into the empty area.

Using the example above, consider the use case of a user wanting to pull the student-worker data from before, but only for males working on Mondays.

1. Drag and drop **Working day** and **sex** into the open area as shown below.
2. Under the **value** header, select Monday, and Male accordingly.
3. Click **Save** then enter preview tab to view and download into Excel.




| | Delete? | Field | Filter Type | Value | Exclude? |
|---|---|---|---|---|---|
| ↕ | | Working Day [CharField] | Contains (case-insensitive) | Monday | |
| ↕ | | sex [CharField] | Contains (case-insensitive) | Male | |


| First Name | Last Name | sex | username | Working Day | team name |
|---|---|---|---|---|---|
| Luke | COLON | Male | lcolon | Monday | test2 |
| Matthew | HOLCOMB | Male | mholcomb | Monday | team 1 |
| Luke | MCCONNELL | Male | lmcconnell | Monday | SuperTeam |
| Christopher | NIELSEN | Male | cnielsen | Monday | President's Team |
| Connor | SANFORD | Male | csanford | Monday | SuperTeam |

## 11.5 Troubleshooting/FAQ

This section is dedicated to helping you troubleshoot your issues with the report builder. Some of the most common user issues are highlighted in bold for your convenience.

**My preview tab won't show results even though the display fields has fields in it.**

The most common reason for this issue is users forgetting to click on save after they have moved the fields they want into the display tab. Keep in mind, the display tab won't show anything unless you have selected save after dragging in your fields.

Example: If you drag in the fields First Name and Last Name then select save you will see those fields in the preview tab. If you go back, drag other fields but never save, when you go to preview, you will only see First Name and Last Name. Selecting save once more will fix the issue and show all your new fields.

**I'm seeing a number of students who are no longer active. I only want active students to show.**

To only see active students, find the *active* field and drag it into your display fields tab. Save. Next, drag the same *active* field into the **Report Filters** tab and set the Filter Type to Equals and the Value box next to it to True. Select Save.

You will now only see a list of students marked as active in the database.

**I can't find the field I'm looking for.**

The first thing you want to check for is to ensure that you're using the correct *root model*. While we provide you with a myriad of root models to choose from, most often you may find what you're looking for by either using the: student, student worker, or applicant root models. From these root models, you can navigate through a number of fields and expanded fields the easiest.

-There's a search box located on the Fields and Expand Related Fields tab- use this to help you locate fields easier.

**The report isn't applying my filters.**

For filters to work, you need to first check that the same field you want to filter is in the display tab. For example, you can't filter by grade level when grade level isn't in the display tab to start off with- in this case, you wouldn't see any change.

You should also make sure that you are using the correct filter type. Typically, the "equals" or "contains" type is the easiest to use, granted whatever you type into the value field is exactly how it appears in the database.

Ex. filter type "equals" set to Sarah will not show any names of Sara. Filter type "contains" will show both.

**Can you utilize multiple values for the same filter?**

Yes. For example, if you have a list of students with their grade levels and you only want to see Juniors and Seniors, then apply the filter type **in(comma separated)** and for value, type in "Junior,Senior" with no space after the comma. The preview will then show only Juniors and Seniors as was requested.

CHAPTER 12

# Code

https://github.com/burke-software/django-sis

CHAPTER 13

# Discussion/Developing Django-SIS

Let us know your thoughts here https://google.com/+Burkesoftware or Contact us! We'd be happy to point you in the right direction.