
Bishop Blanchet Intranet Documentation

Release 1.0

Luis Naranjo

December 11, 2013

Contents

1	What is it?	1
2	LDAP Authentication	3
3	Types of users	5
3.1	Super user	5
3.2	Regular user	5
4	Permissions	7
4.1	Can sign up other chaperones	7
4.2	Can remove other chaperones	7
4.3	Can view other chaperones	7
4.4	Can sign up self from an event	7
4.5	Can unsign up self from an event	8
4.6	Can create events	8
4.7	Pull user reports	8
5	Groups	9
5.1	Staff	9
5.2	Chaperone_Requirement	10
5.3	Chaperone_Site_Admin	10
5.4	Intranet_Admin_Access	10
5.5	Intranet_Super_Admin	10
5.6	Chaperone_Event_Manager	10
6	Scripts	11
6.1	syncUser.py	11
6.2	syncGroup.py	11
6.3	reset.py	11
6.4	manual_reset_db.sh	12
6.5	Other	12
7	Chaperone App	13
7.1	Event listings page and Event pages	13

7.2	Types of users in the chaperone app	13
7.3	Markdown rendering for event descriptions	14
7.4	Yearly Deadline	14
7.5	Emails	15
8	What to do if:	17
8.1	A user can't sign in	17
8.2	You want to find an old event	17
8.3	You want to pull up random info	17
8.4	You want to disable the email reminders	18
8.5	You need to migrate the site	18
8.6	You need to reset the database	18
8.7	You want to change the default event requirement number	18
8.8	You want to delete a user	19
8.9	You want to send an email advertisement for an event	19
8.10	You want to change the year end and year start reset dates	19
8.11	You want to change LDAP settings	19
8.12	You messed up/deleted the code	19
8.13	You want to modify or change HTML	19
8.14	You want to add static files	20
8.15	Error: DoesNotExist at /chaperone/addEvent/	20
8.16	What happens to people who miss the deadline?	20
9	Indices and tables	21

What is it?

The intranet is an internal web service designed for use by the Bishop Blanchet Staff.

Any BBHS faculty member or BBHS school administrator can log in and use the site with their existing school username and password.

The idea is that within the intranet, there are several “apps” that can be used by staff members. Right now, we only have a chaperone app, but we are looking at expanding the intranet with more apps in the future.

LDAP Authentication

The intranet makes extensive use of the BBHS Active Directory for user authentication.

It's important to note that the user that is being used by the site for binding is:

```
'CN=Luis Naranjo,OU=Technology,OU=Staff,DC=campus,DC=bishopblanchet,DC=org'
```

and the password is a certain type of baked goods that happened to be the default password in the first Tech Apprentice class.

So don't move that user, change his password, or change his permissions unless you change them in the code too.

If you do change them in the code, you would have to modify the `~/bbhs_intranet/bbhs/settings.py` file.

Make sure you restart the apache2 service.

Types of users

There are two main types of users in the intranet.

Every user is either a super user, or a regular user.

All super users have the same full permissions, but not all regular users get the same permissions. That depends on which user-specific permissions they have been explicitly granted, or which permission groups they belong to.

3.1 Super user

A super user has full permissions to access the admin page, and they can add, remove, or modify any object in the database.

Additionally, a super user has full permissions to do any app-specific actions.

3.2 Regular user

Globally, a regular user only has permission to login to the intranet with their LDAP credentials. The only people who can login are those that are included in the Staff security group.

A regular user may have further bare-minimum permissions that are specific to the apps of the intranet.

Permissions

By default, Django comes bundles with a bunch of different permissions that users and groups can receive (if a user is a member of a group, that user gets all of the group's permissions added to his/her own).

These permissions are all related to editing, adding, and deleting objects in the database through the admin page.

They look like this:

```
admin | log entry | can add log entry admin | log entry | can change log entry admin | log entry | can delete log entry
```

In addition to these standard permissions, I've added permissions that are specific to each application.

Here are the relevant permissions for the chaperone app:

4.1 Can sign up other chaperones

This allows users to sign up other users at an event's page.

This should not be given to the regular user. Maybe no one should be able to do this except the superusers who have this permission implicitly.

4.2 Can remove other chaperones

This allows users to unsign up other users at an event's page.

4.3 Can view other chaperones

This permissions allows users to see what other users have signed up for an event.

By default, everyone can see how many volunteers are still needed, but to see who is signed up you need this.

4.4 Can sign up self from an event

Every regular user should have this. Without this, they can't sign up for events

4.5 Can unsign up self from an event

Without this, they can't unsign up for events

4.6 Can create events

Give this to whoever might be creating events.

This allows them to access the /addEvent page

4.7 Pull user reports

This permission allows users users to pull user reports on other users in the chaperone app

Groups

The system is set up to be flexible and easily customizable. The site primarily uses groups to handle user's permissions. I tried to set it up so the group changes are primarily changed in the existing LDAP database.

In addition to using groups to handle user's permissions, individual user permissions can be modified at will in the admin page with the same effect.

It's important to note that the site will mirror all of the user's LDAP groups over to Django if the groups are within the "Staff" ou. There will probably a few extraneous groups out there and should be ignored.

Also, the mirroring system over-writes local permissions with LDAP ones when a user logs in. For example, say a user is not part of the event admin LDAP group, but is made part of the event admin DJANGO group by a super user. The user will show up as an option as an Event Admin in the Add an Event page, and for all practical purposes will be an Event Admin - until his/her account is synced with LDAP on login. When that happens, the user will be stripped of event admin permissions on the site until he/she is actually added to the correct group on the LDAP end and the site is synced with LDAP.

Moral of the story, make changes on the LDAP end first. After you make a change, get the user to log in for the changes to take effect OR manually go in to the admin page and update the user's permissions (they will still be over-written on log in, but there won't be any change) OR run the sync scripts i've provided.

There are 5 important security groups to consider.

The following groups **inherently** grant user's specific abilities *just by being members of them*. These abilities are described in their corresponding sections below.

But if you click on the groups in the admin page and look at the permissions that they propagate, you'll see that some by default they grant none. You can customize what additional permissions users can get here.

For example, the Intranet_Admin_Access group just grants users the ability to log in to the admin page. Users who get this ability will probably want the ability to edit certain objects in the database too, and this is the kind of permission that can be customized here.

5.1 Staff

```
cn=Staff,ou=staff,dc=campus,dc=bishopblanchet,dc=org
```

This security group contains all of BBHS's staff. It is also the requirement for authentication. If a user doesn't belong to this security group, the user can't log in even if they have the correct username and password.

All members that authenticate via LDAP will be members of this group.

All members of this group get permission to sign up, and unsign up from events.

5.2 Chaperone_Requirement

cn=Chaperone_Requirement,ou=Intranet,ou=Technology,ou=Staff,dc=campus,dc=bishopblanchet,dc=org

A user will not get any monthly email reminders of their chaperone obligation unless they are a part of this group. This security group needs to be populated with all of the teachers.

5.3 Chaperone_Site_Admin

Members of this group will be granted elevated permission to add, modify, and delete any Chaperone objects in the database.

They will also get the ability to pull user reports.

In other words they have permission to do anything that is related to the chaperone app.

Additionally, they can edit user info in the admin page such as change a user's event unit requirement.

Joan Thompson is probably the only person who needs this access.

Make sure to add her to the Intranet_Admin_Access so she can access the admin page.

5.4 Intranet_Admin_Access

cn=Intranet_Admin_Access,ou=Intranet,ou=Technology,ou=Staff,dc=campus,dc=bishopblanchet,dc=org

Joining this group grants users the permission to log in to the admin page at <http://faculty.bishpoblanchet.org/admin/>, but nothing more (by default).

5.5 Intranet_Super_Admin

cn=Intranet_Super_Admin,ou=Intranet,ou=Technology,ou=staff,dc=campus,dc=bishopblanchet,dc=org

Members of this group get full global permission to do anything in the site.

The only exception to the unfettered power that comes from being a member of this group is the ability to log in to the admin page (which is essential for a “super user”). To be able to do that, **a super user also needs to be a member of the Intranet_Admin_Access group.**

5.6 Chaperone_Event_Manager

cn=Chaperone_Event_Manager,ou=Intranet,ou=Technology,ou=Staff,dc=campus,dc=bishopblanchet,dc=org

Joining this group allows users to be Event Administrators. All members of this group will show up in the dropdown menu for Event Administrator in the “Add a new Event” page.

Scripts

I wrote a few administrative scripts that will come in handy. They are located in `~/bbhs_intranet/scripts/`

6.1 syncUser.py

This script will synchronize the selected user's information from the active directory with the intranet database.

Run this script immediately after making changes to the LDAP groups in order to apply the changes immediately on the intranet side.

If you don't, then the changes won't happen until the user who has had his/her permissions modified next logs in.

For some reason, the AD is a bit laggy in updating information and it sometimes won't update it's query responses for about 30 seconds after the change.

Make sure the changes have in fact taken place, and if not, then keep running this script till they do.

Call this script like so:

```
$ python syncUser.py luisadmin
```

6.2 syncGroup.py

This script does the exact same thing as `syncUser.py`, but it is designed to be used for groups instead of individual users.

You can sync ou's and security groups with this script.

Just run it and follow the instructions it gives you.

6.3 reset.py

This is a python script I wrote for resetting certain parts of the website.

For example, say you want to erase all of the events in the database, or all of the users, etc...

6.4 manual_reset_db.sh

This script is a hard reset.

It actually deletes the database and creates a new one.

Only use this if the database is somehow totally hosed and you don't mind losing your data.

6.5 Other

There are a few other scripts here that were used for development and/or are obsolete.

They won't do any harm, but don't use them anyways.

They are:

- defaults.py
- permissions.py
- popFaculty.py

Chaperone App

The chaperone app was created to replace an existing system for managing school events and their volunteers.

7.1 Event listings page and Event pages

There is an event listings page that lists all of the future events in order of which one is most close to taking place. Users can click on an event and be taken to that event's corresponding page.

There, users will be able to see things like the event's date, time, description, administrator, administrator contact information, notes from the administrator or from other volunteers, a list of other volunteers, contact info of other volunteers, and how many volunteers are still needed.

Additionally, users can sign up or unsign up for that particular event on that page.

7.2 Types of users in the chaperone app

There are 2 main types of regular users in this app:

- Event Administrators
- Chaperones

7.2.1 Site Administrators

These are the people in the activities office who are going to be creating the events and designating who is the event administrator.

7.2.2 Event Administrators

Each event has a single **event administrator**, that is set initially by the site administrator that created the event. This can be changed by a site admin later.

There is a restricted group of users who are allowed to be event administrators.

Any site administrator can add or remove users from this group.

This is a Django group and LDAP Security group, and it is called “Intranet_Event_Admin”. If the database is ever reset, this group needs to be re-created manually through the admin page verbatim or a user who belongs to this security group needs to log in. If not, the “Add a new event” page will not work

The event admin’s contact info will be posted on the event’s description page, as well as a list of all of the other users who have signed up for the event. This is safe, because only logged-in users can see this information.

Event administrators can

- Send emails to the whole BBHS faculty to request volunteers (through a site admin)
- Send email reminders to the event’s signed up users (through a site admin)
- Read the public notes and the private notes that users post on the event’s page
- Sign up other users directly for the event
- Remove other volunteers from the event’s registration

7.2.3 Regular users

Most users in the system will be placed here. The regular user will be able to view all future events, and will be able to sign up or unsign up for events.

The events have a weight system. Every teacher has a fixed yearly requirement. At the time of writing, that requirement was 4 event units per year. Each event has it’s own weight. For example, Kairos might be worth three event units since it takes place over three days.

Every time a user signs up for an event, his or her required chaperone event count will be decreased by that event’s corresponding weight. Likewise, when a user unsigns up, the count will increment by that weight.

If a user goes past his or her requirement count, it will appear as if the user has reached a plateau of 0 required events, but the application is behind the scenes going into negative numbers in case that data is ever desired.

Additionally, every time a user signs up for or unsigns up for an event, the action, date and time are logged in the database.

7.3 Markdown rendering for event descriptions

When a user adds an event with the custom form, they get the option to use markdown rendering.

Markdown is a powerful text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML).

This document that I wrote is written using a similar text-to-HTML conversion tool called restructured text. It’s more confusing, don’t look at the source. Check out the links below instead.

Here is a good example article written in markdown:

<http://daringfireball.net/projects/markdown/>

A cheat sheet:

<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

7.4 Yearly Deadline

On the deadline date, it will check for those who missed the deadline and let the interested parties know.

At the beginning of the school year, it will reset all the yearly counts (events done so far, events still needed).

The correct dates need to be updated in `~/bbhs_intranet/bbhs/settings.py` at the very bottom. The years need to be incremented by 1 every year.

It will send the ADMINS in the settings.py file email reminders when this is necessary.

7.5 Emails

The chaperone system will send out a variety of emails automatically.

Every month, it will go through the system and send an email reminder to all of the members of the Intranet_Chaperone security group that still haven't fulfilled their requirement.

Every day, it will go through all of the events that are happening with 2 days, and let the registered volunteers have an email reminder

When a user signs up for an event, it will email the event administrator letting them know who signed up, and it will send the user who signed up an email confirmation.

Once a year, it will send an email to the site admins (Michael, David, Luis) with instructions on how to update the start school date and the end school date so it knows when the chaperone deadline is, and when to reset the user's yearly event counts.

Once a year, it will also send an email to the designated `DEADLINE_EMAIL` variable defined in `~/bbhs_intranet/bbhs/settings.py` letting them know what users missed the deadline for required events.

What to do if:

Here are some pre-emptively formulated FAQ's for the admins

8.1 A user can't sign in

Check that they are in the Staff security group Check that their account hasn't been disabled by an admin

8.2 You want to find an old event

Go in the admin page and look for them there

8.3 You want to pull up random info

Say you want some information that isn't visible on the page, but you are pretty sure is stored in the database somewhere. Give me a call and I can do some database queries for you.

For example, say you want a list of all the members of group "x", or you want to see a user's sign up/unsign up history, or you want to see a custom user report such as how many events does user a have in common with user b.

Those are just a few of many things that can be quickly computed with django's database query API in python. Feel free to take a look at the source code to get a feel for what data is in there and what's not.

<https://github.com/luisnaranjo733/bbhs-intranet>

The database stuff is in chaperone/models.py or intranet/models.py

If you want to hack at it yourself, here is a sample query:

```
cd ~/bbhs-intranet python manage.py shell from chaperone.models import Event, Note from intranet.models import UserProfile
```

```
a = UserProfile.objects.get(username='cconnors').user b = UserProfile.objects.get(username='mfreyman').user
```

```
matches = []
```

```
for event in Event.objects.all(): volunteers = event.getVolunteers() #this is a list of volunteers if a in volunteers and  
    b in volunteers:
```

```
    matches.append(event)
print matches
```

8.4 You want to disable the email reminders

Just disable the cron jobs:

```
~/bbhs_intranet/chaperone/cron/daily.py
```

This job runs once a day, and it checks for events that are happening within two days.

It sends an email reminder about the event to all of the signed up chaperones.

```
~/bbhs_intranet/chaperone/cron/daily.py
```

This runs once a month, and it goes through the users and checks to see if they've fulfilled their service requirement. If they haven't, then it sends them an email to remind them to get on it.

8.5 You need to migrate the site

It's a pain in the ass and I don't advise it.

```
sudo apt-get install git git clone https://github.com/luisnaranjo733/bbhs-intranet //will download code
```

```
sudo apt-get install python-pip // installing dependencies for python2.7 sudo pip install -r requirements.txt
```

You need to set up a dedicated python production server. I used mod_wsgi for apache. There are nginx options and several other python specific servers.

```
sudo mv ~/bbhs-intranet/faculty.bishopblanchet.org /var/www/faculty.bishopblanchet.org you may have permission issues with this - fix with chmod
```

```
sudo apt-get install ldapsearch
```

8.6 You need to reset the database

```
cd ~/bbhs-intranet sudo sh reset_db.sh // log in with someone who is a member of the "Staff" security group or // manually create a group named "Staff" in the admin page python populateFromLDAP.py // this will populate everyone in ou=Faculty python populateFromLDAP.py staff // this will populate everyone in ou=Administration
```

8.7 You want to change the default event requirement number

```
edit ~/bbhs_intranet/bbhs/settings.py
```

Change OBLIGATION_NUMBER = 4 Go to the admin page, then go to Users Select the admin action "Normalize Yearly Obligation" and select everyone

This will reset every active user's events done count for this year, and their events needed count for this year.

8.8 You want to delete a user

You should probably just disable them. Go to the admin page and click disable and then save.

I coded it so that disabled users don't get to do anything.

8.9 You want to send an email advertisement for an event

Go to the admin page, go to events, and select all of the events you want to advertise, then select the "event ad for unregistered users" action and hit go.

Select the "event reminder for registered users" if that's what suits you

8.10 You want to change the year end and year start reset dates

Edit the very bottom of the settings file

8.11 You want to change LDAP settings

Just modify the settings.py file. It's pretty straightforward, but let me know if you need any help. All relevant LDAP related variables are prefixed with AUTH_LDAP_*

Make sure to restart the apache server for changes to take effect

8.12 You messed up/deleted the code

```
cd ~/bbhs_intranet/ git checkout - .
```

This will reset all the code in this directory to the state I left it in last.

This will not change /var/www/faculty.bishopblanchet.org, so I recommend not touching that folder

In case you deleted ~/bbhs_intranet or ~/bbhs-intranet/.git

```
git clone https://github.com/luisnaranjo733/bbhs-intranet ~/bbhs_intranet
```

8.13 You want to modify or change HTML

The html templates are located in ~/bbhs_intranet/bbhs/templates

You can safely make changes, but make sure you don't delete anything that looks like:

```
{% stuff %} // template engine constructs, for loops, block tags, etc.. or {{ stuff }} variables
```

If you want to get fancy look up Django's templating system. It's not hard to learn.

You could take a look at chaperone/views.py or intranet/views.py file for figuring out what objects and lists are being passed to the templates before they are rendered in static html

8.14 You want to add static files

Add your static files to `~/bbhs_intranet/bbhs/static/`

```
cd ~/bbhs_intranet/ sudo python manage.py collectstatic //say yes // you may have to restart the apache server
```

8.15 Error: DoesNotExist at /chaperone/addEvent/

Group matching query does not exist. Lookup parameters were { 'name': 'Intranet_Event_Admin' }

Create the `Intranet_Event_Admin` group in the admin page or log in with a user who is a part of that group

8.16 What happens to people who miss the deadline?

There will be an email that will go out to whoever the `DEADLINE_EMAIL` variable is set to in `~/bbhs_intranet/bbhs/settings.py` that will contain a list of the users who missed the chaperone requirement deadline, and their information.

Developed by [Luis Naranjo](#) 2013

Source code: <https://github.com/luisnaranjo733/bbhs-intranet>

Indices and tables

- *genindex*
- *modindex*
- *search*