**HIPAA Squad 2.0 Project Documentation**

Erin George, Jeremy Clark, Ricky Renner, KC Miller

**Vision and Scope**

**Vision Statement:**

HIPAA Squad 2.0’s vision is to have a software that doctors can use to refer patients to the OSU Neurosurgical Department and set up appointments through the online portal instead of through paper and pen scheduling.

**Scope:**

The scope of this project is to make progress on the existing project and to bring it closer to being able to be deployed. It is not expected that the team will have a fully functional web application, and the team should not worry about hosting it anywhere expect locally on their computers.

### **Actor’s Catalog**

**Site Admin**

* Can create/edit/delete users in the system.
* Can create/edit/delete clinics in the system.
* Can create/delete specializations in the system.
* Can create/delete conditions for specializations in the system.
* Can modify their own account.

**Referring Doctor**

* Can search/filter through all of the referred doctors in the system.
* Can create referrals in the system.
* Can view referrals that are pending approval.
* Can modify their own account.

**Referred Doctor**

* Can view/accept/deny referrals for them.
* Can modify their own account.

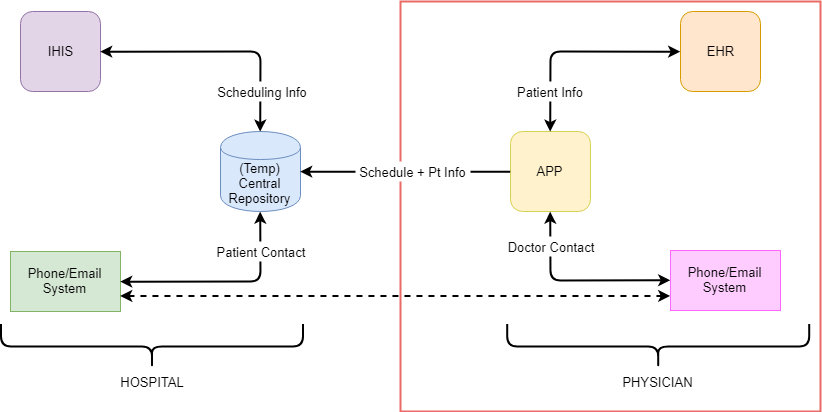
**Referring Staff**

* Can search/filter through all of the referred doctors in the system.
* Can create referrals in the system.
* Can view referrals that are pending approval.
* Can modify their own account.

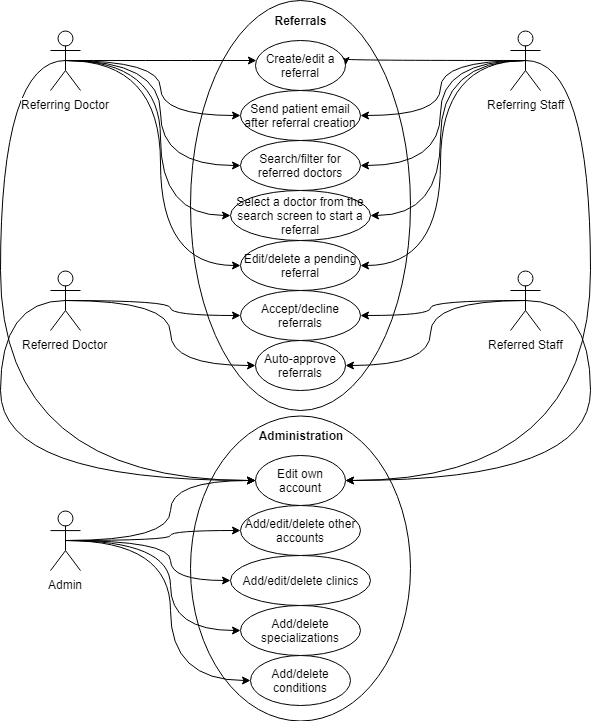
**Referred Staff**

* Can view/accept/deny referrals for their clinic’s doctors.
* Can modify their own account.

**Architecture and Design**



**Use Case Diagram**

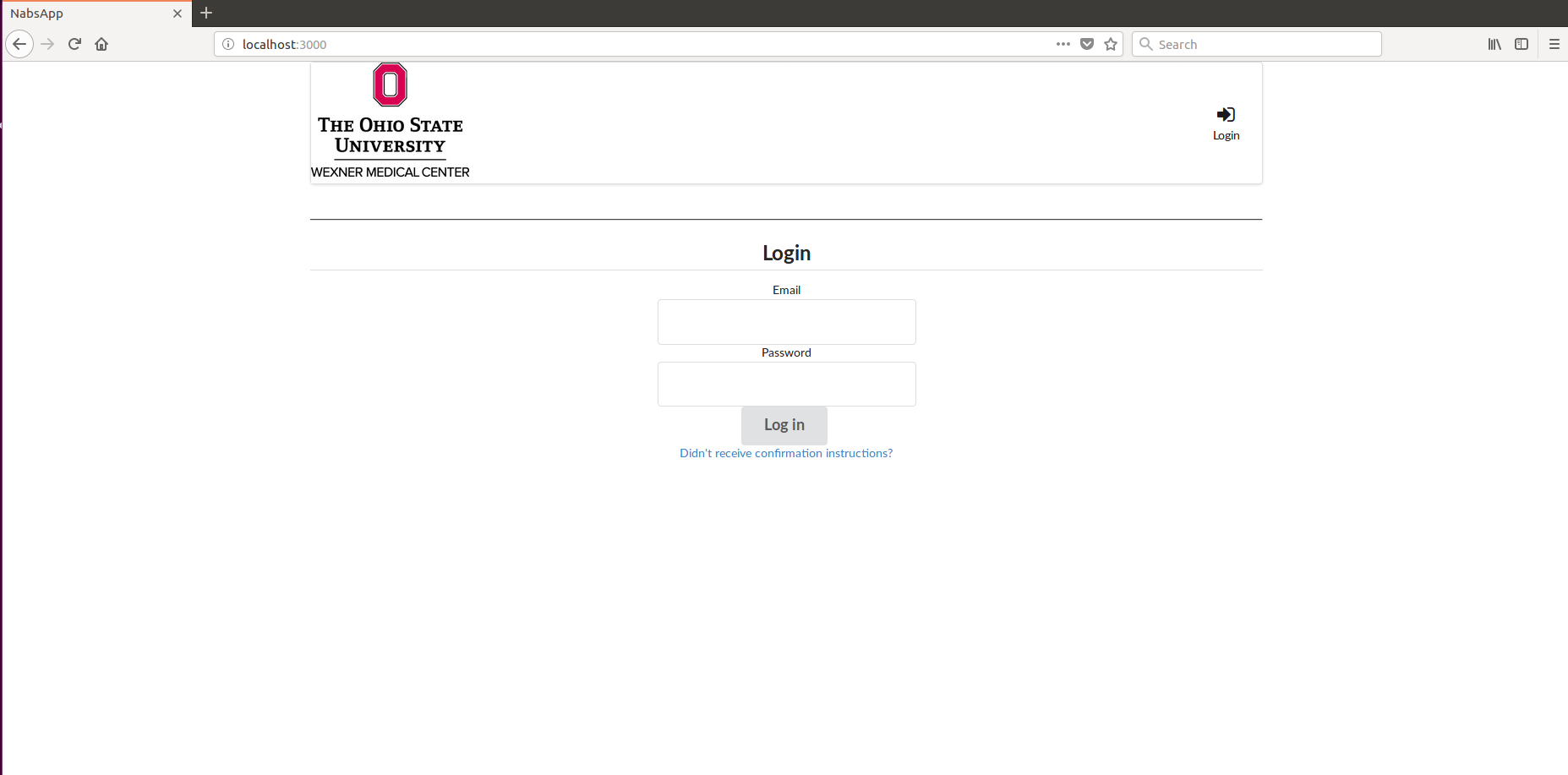


**Use Cases**

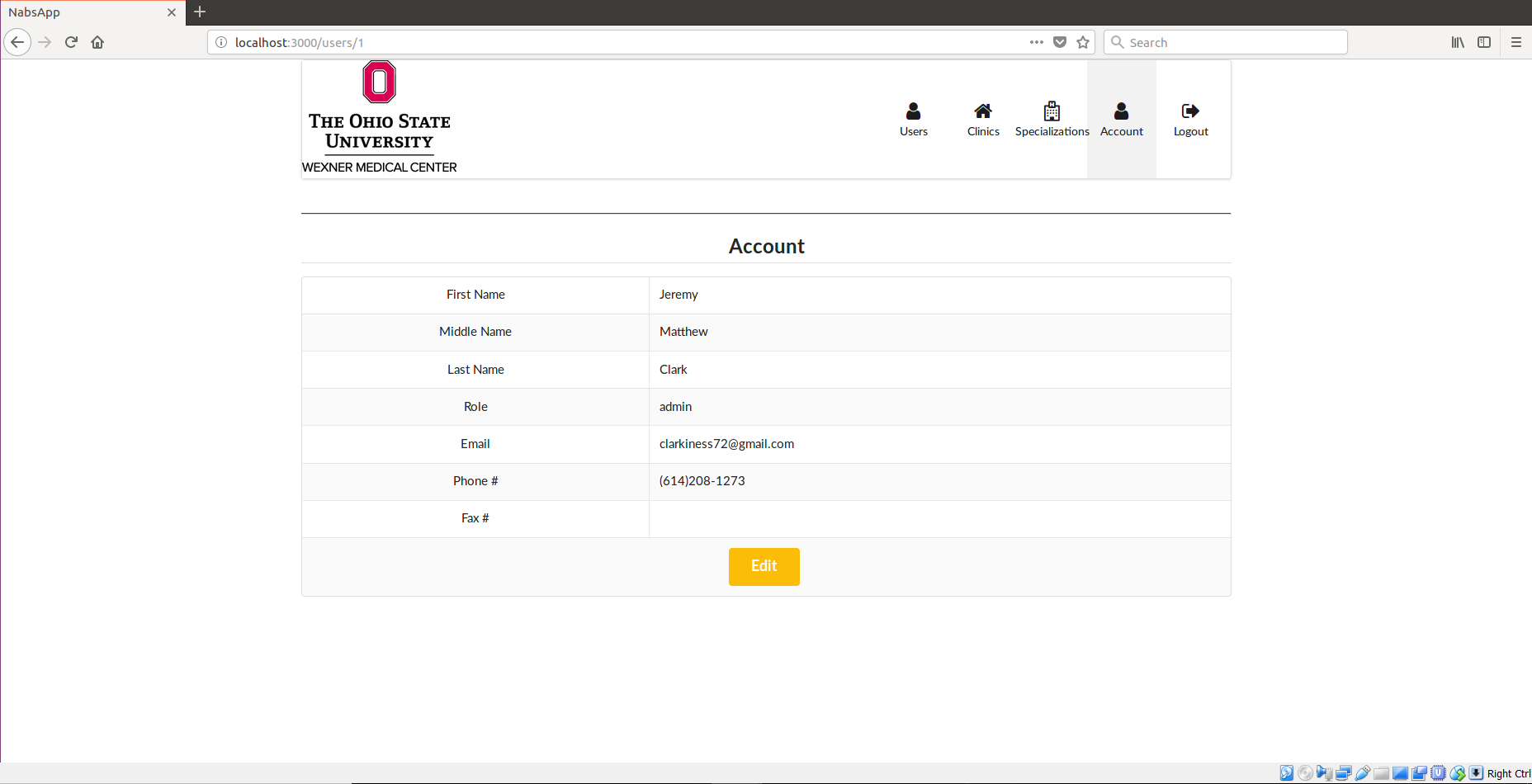
* As a doctor outside of OSUMC, I want to be able to log into my account so that I can use the scheduling app and see my referrals.
* As an OSUMC doctor, I want to accept an appointment request so that I can have the appointment set up on my calendar.
* As a doctor outside of OSUMC, I want to choose a specific date to schedule on so that my patient get a date that works with their schedule.
* As a doctor outside of OSUMC, I want to choose a specific doctor to schedule with so that my patient can build a relationship with the doctor every appointment.
* As a doctor or a patient, I want to receive an email confirmation of my appointment so I know when and where the scheduled meeting is.
* As a doctor, I want to be able to have referrals be auto-approved in order to save myself the time of having to approve every referral.

**User Guide**

**All Users:**

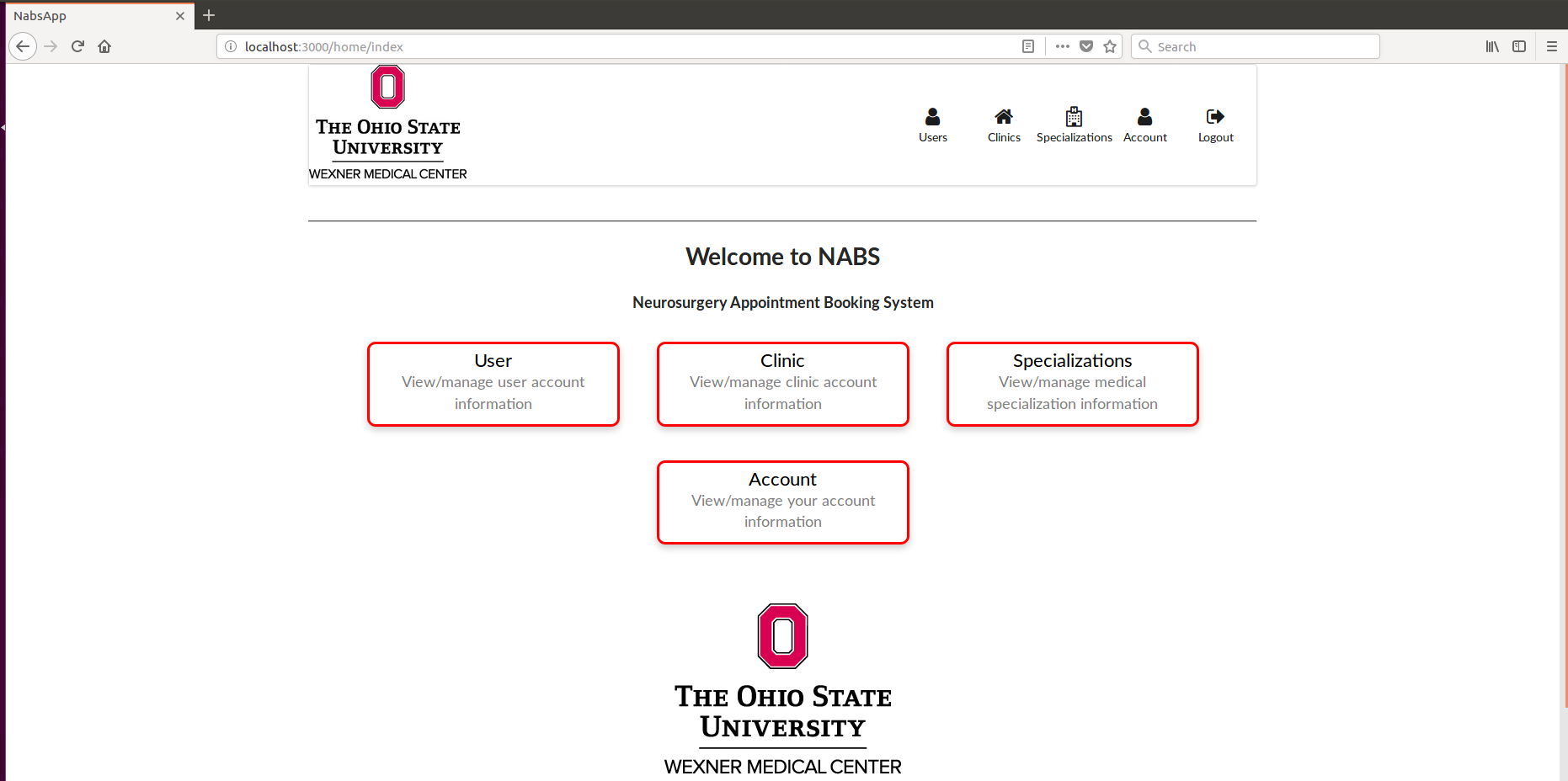


All users must login with their email and password. The account will be set up by the site admin creating your account followed by a set up email which will give you an initial password and a link to follow to finish setting it up.

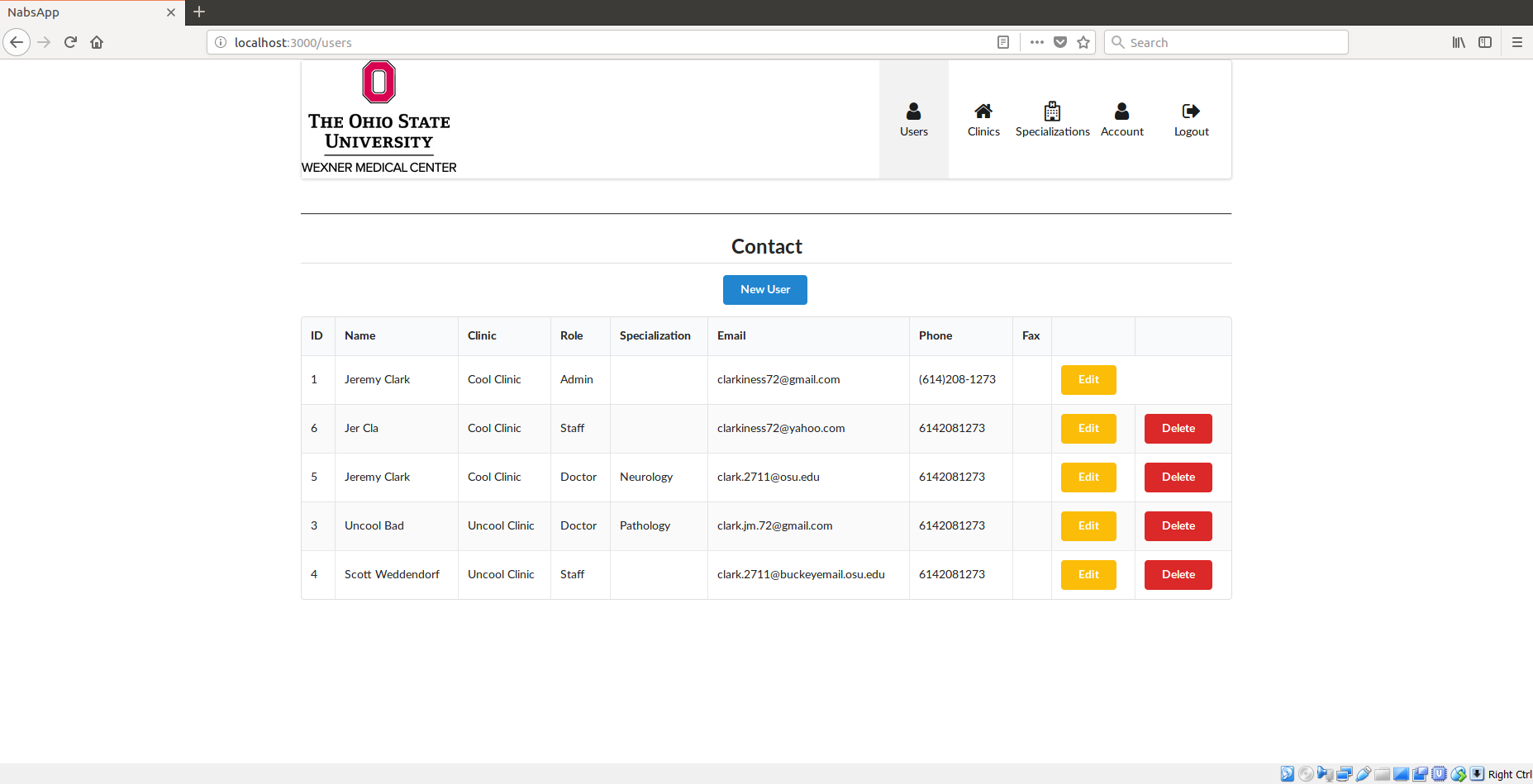


All users are able to view/modify their own accounts by going to the Account page.

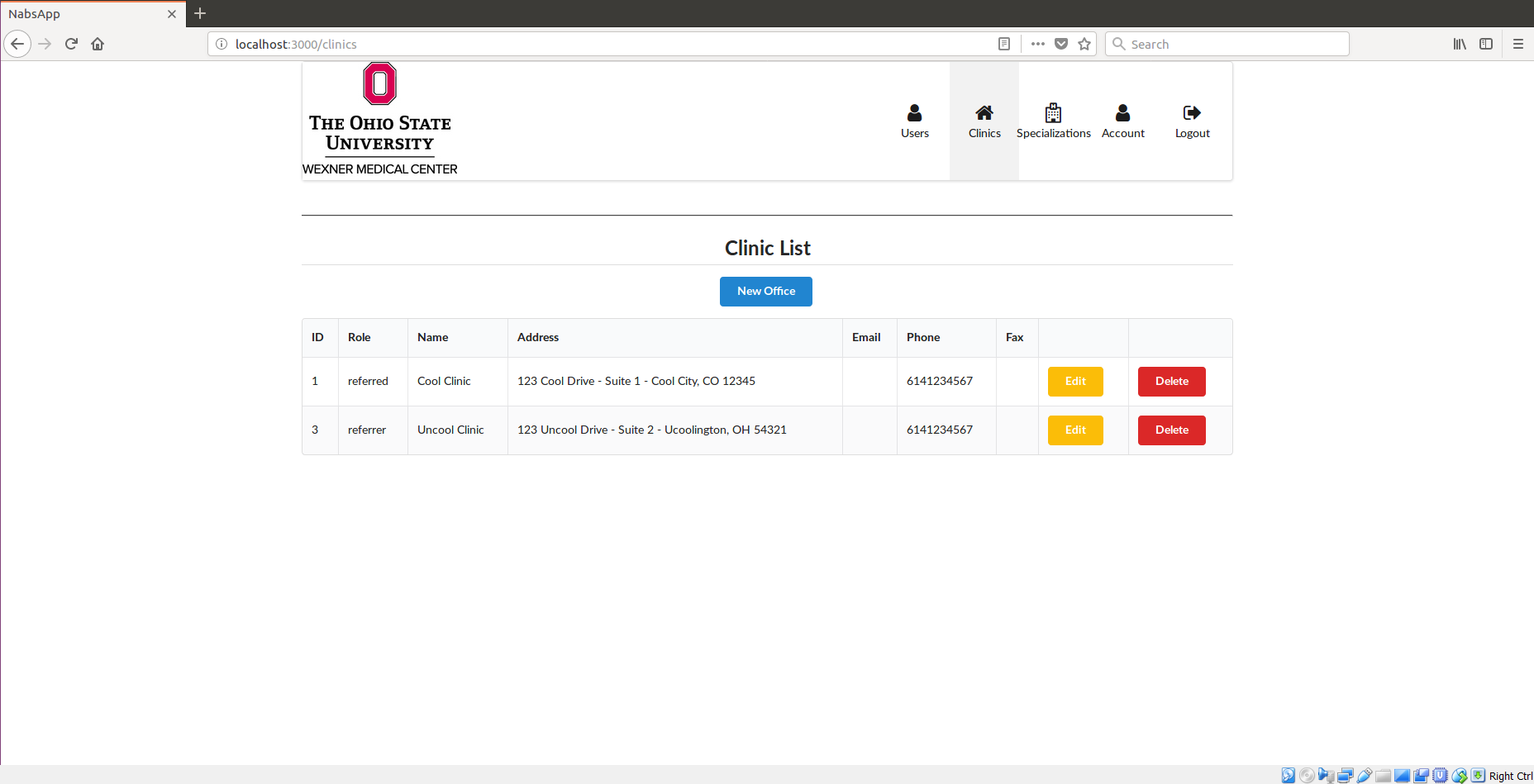
**Admin:**



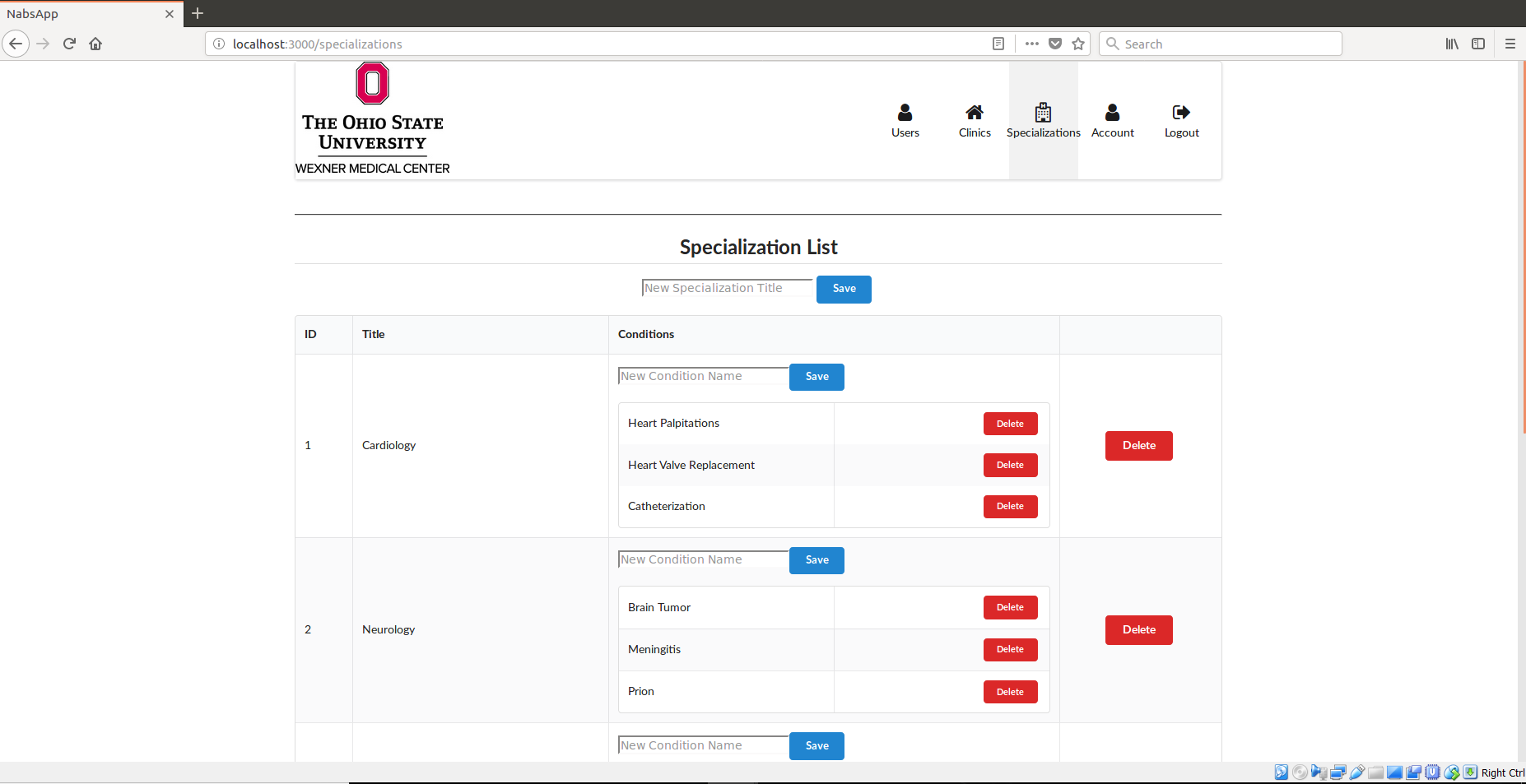
Admin users are able to manage all users, clinics, specializations, and their own account.



Admin users are responsible for managing all of the users within the system. Admin users must manually add users to the system for them to be able to access the system.

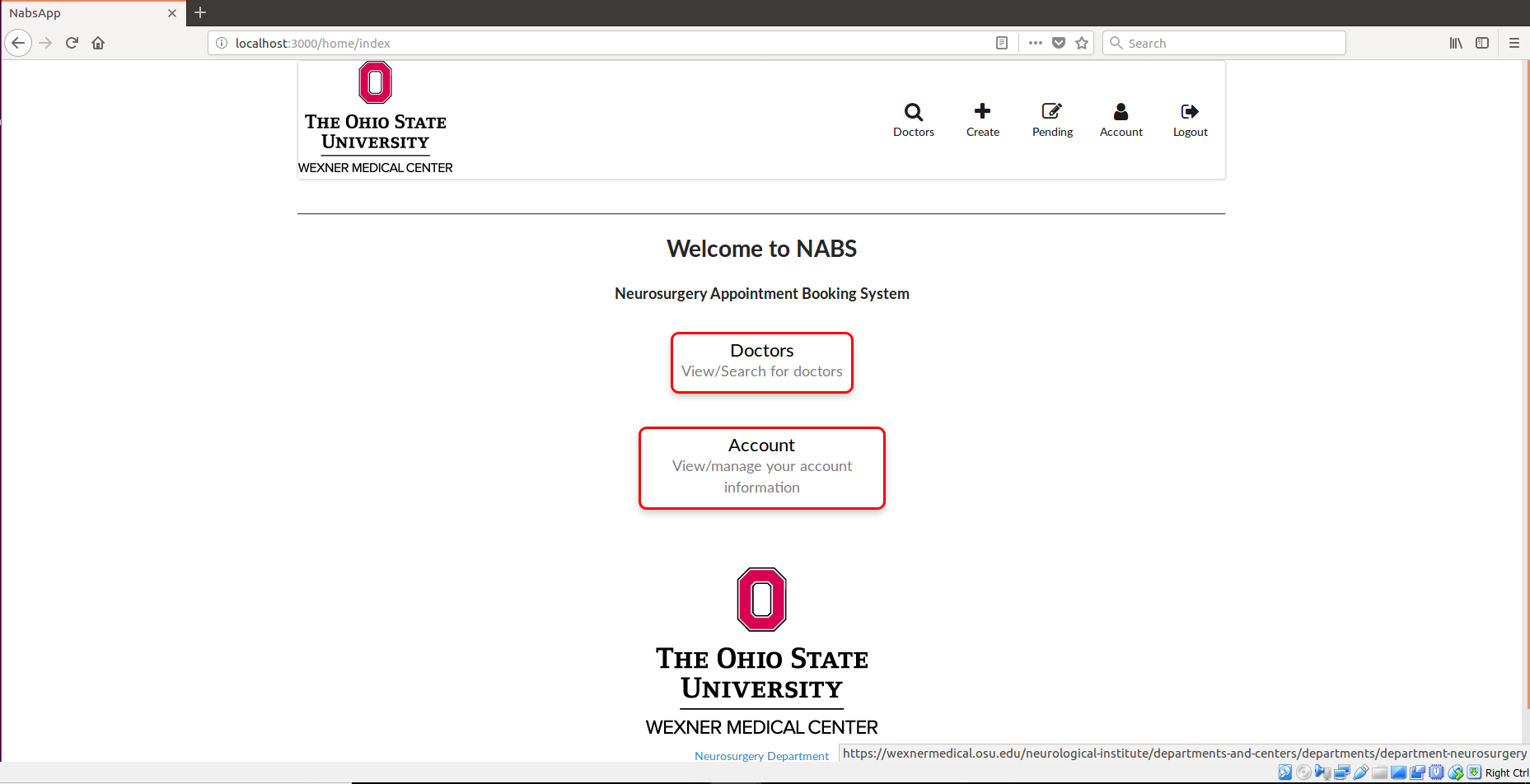


Admin users are responsible for managing the clinics in the system.

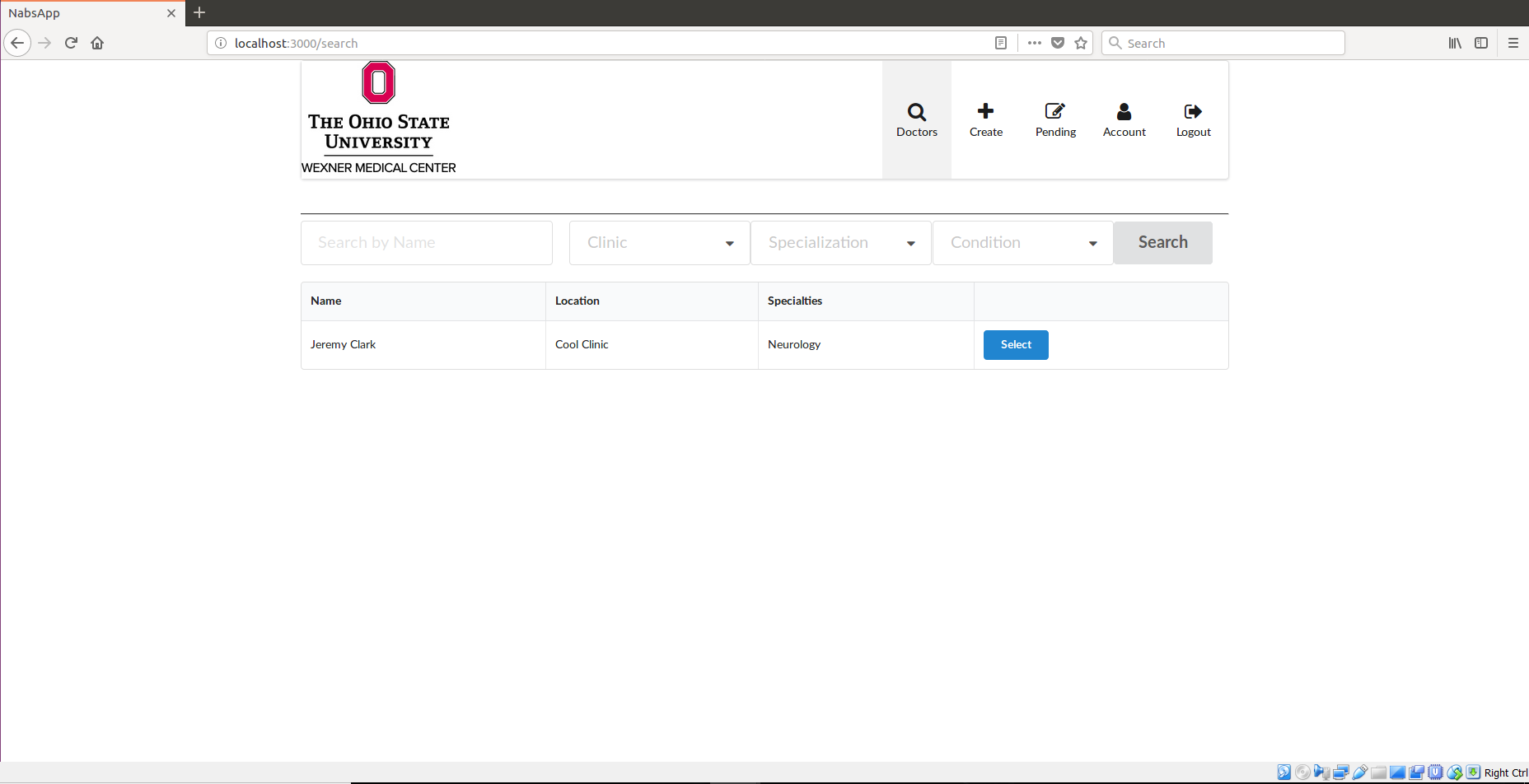


Admin users are responsible for managing the various specializations in the system.

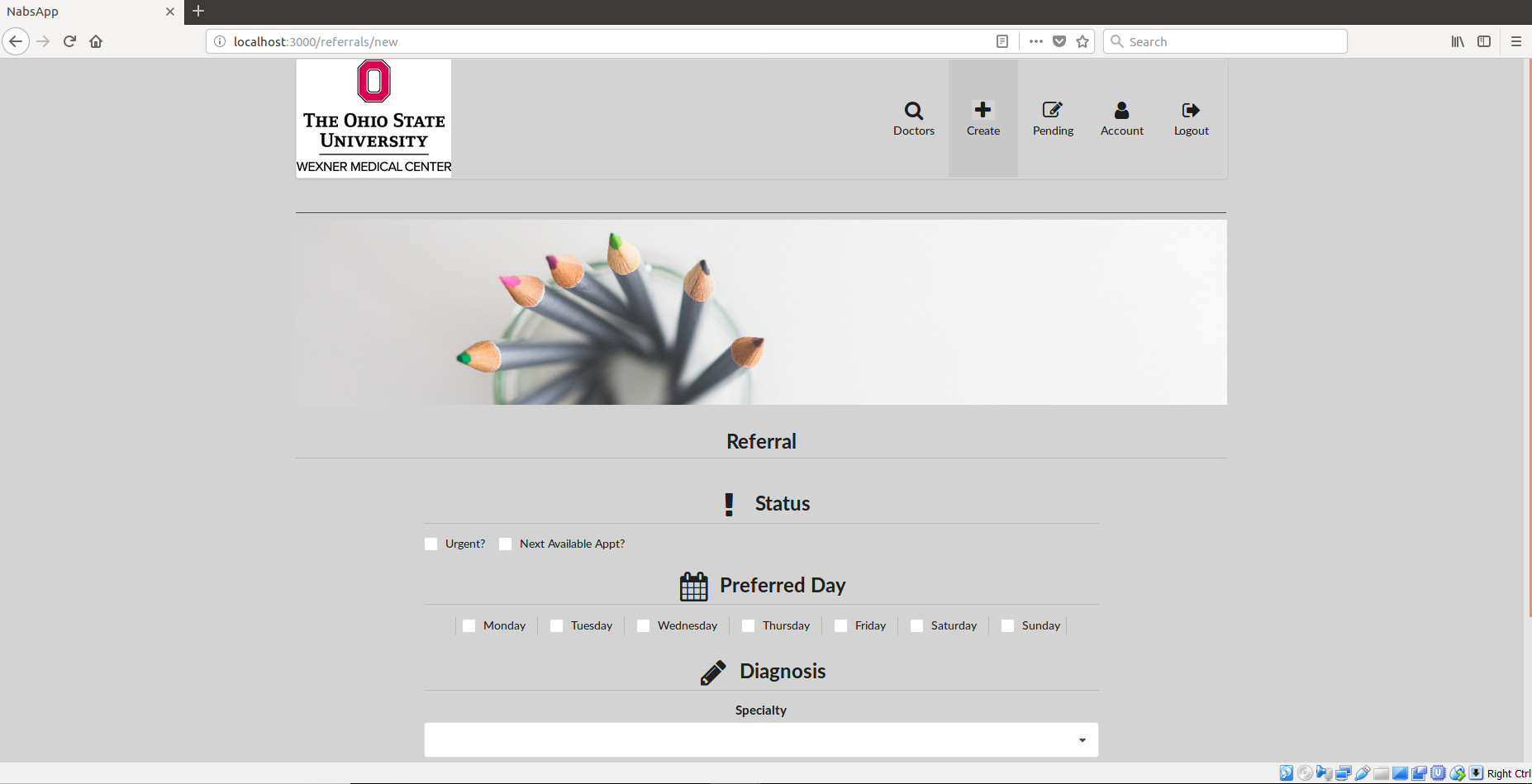
**Referring Doctor and staff:**



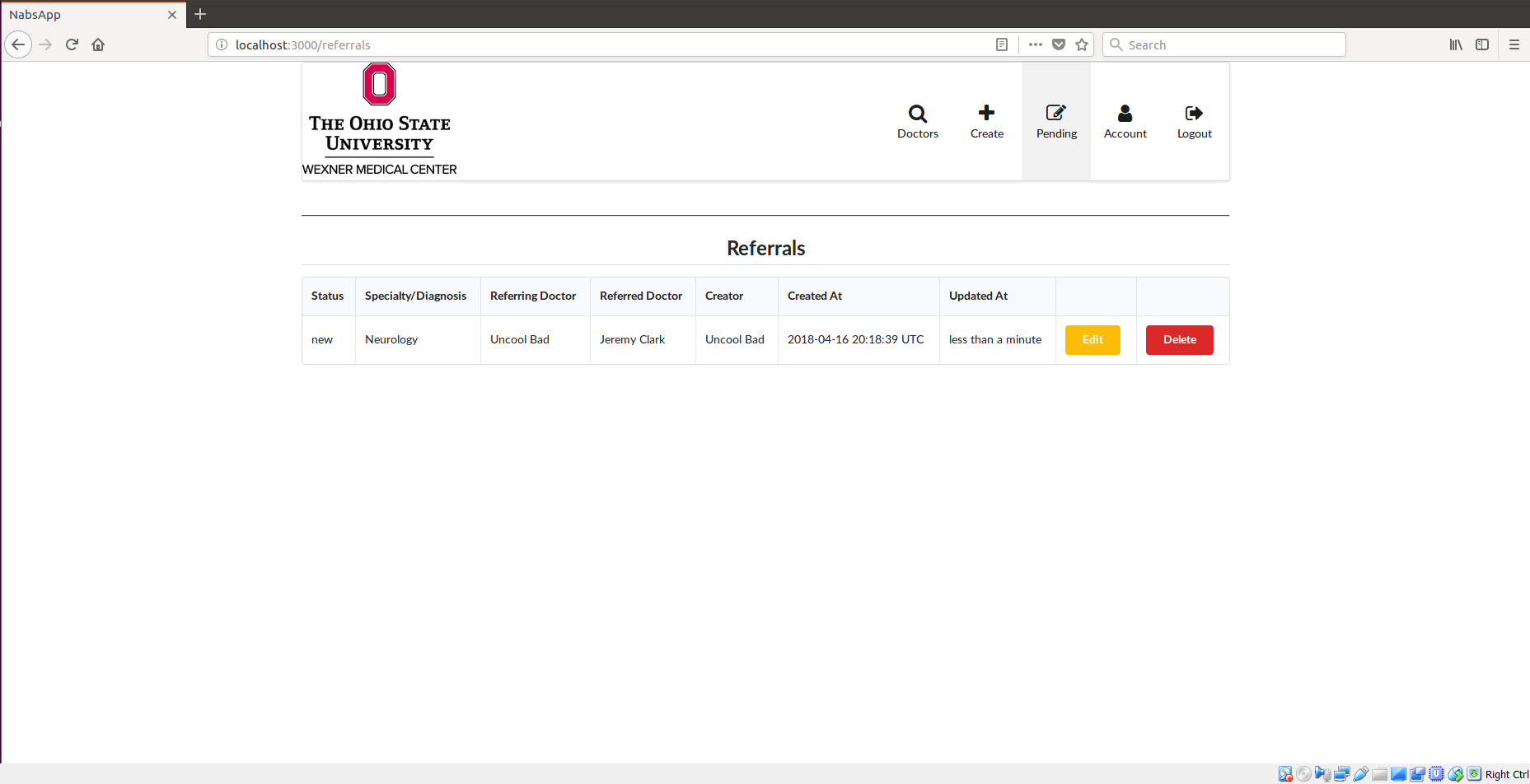
The referring doctors and staff are able to search through the doctors in the system, create referrals, edit pending referrals, and manage their own accounts.



Referring doctors and staff can search for referred doctors by name, clinic, specialization, and conditions treated. The select button will start a referral for that doctor and bring you to the referral creation page.

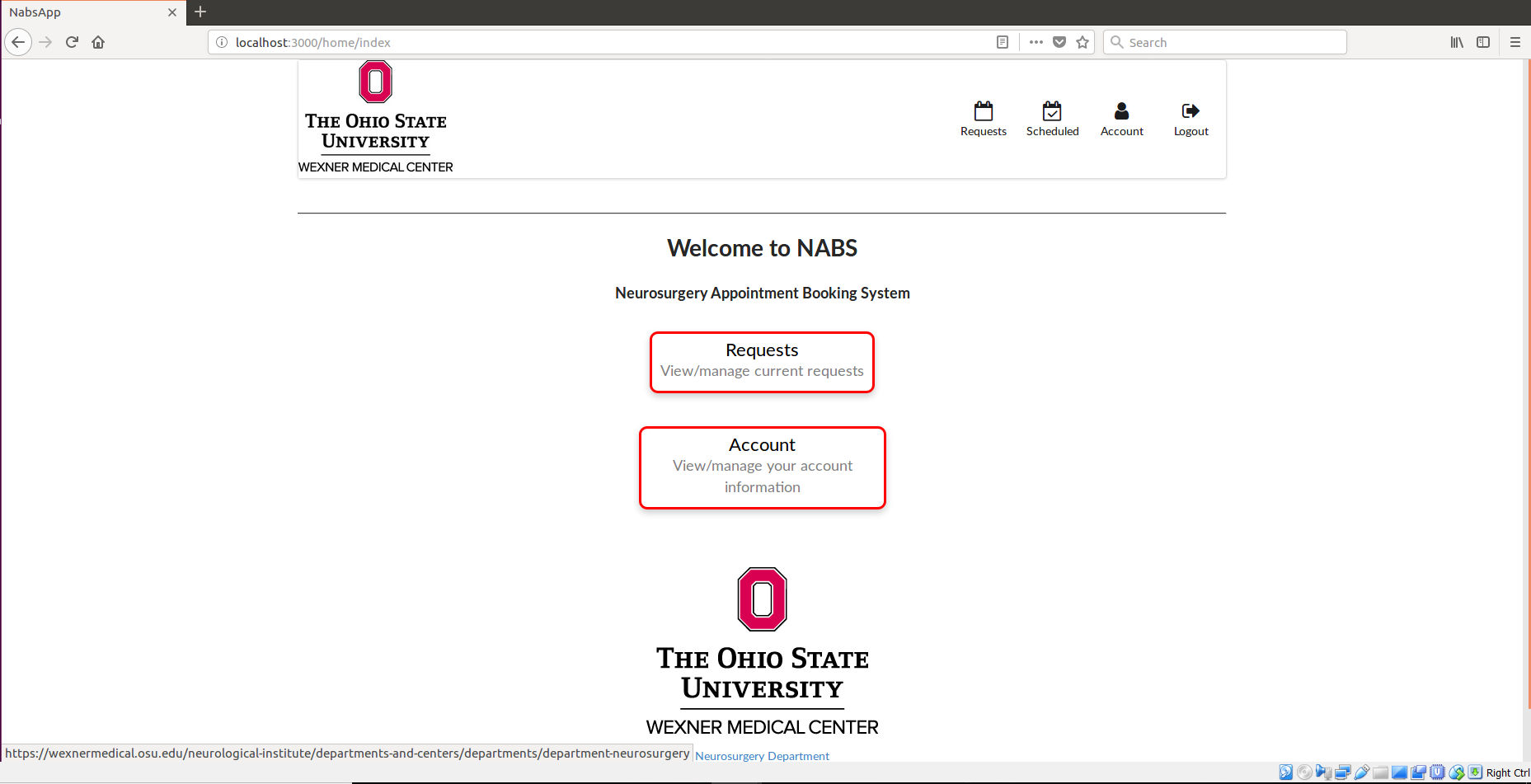


Referring doctors and staff can create referrals for a doctor in a referred clinic.

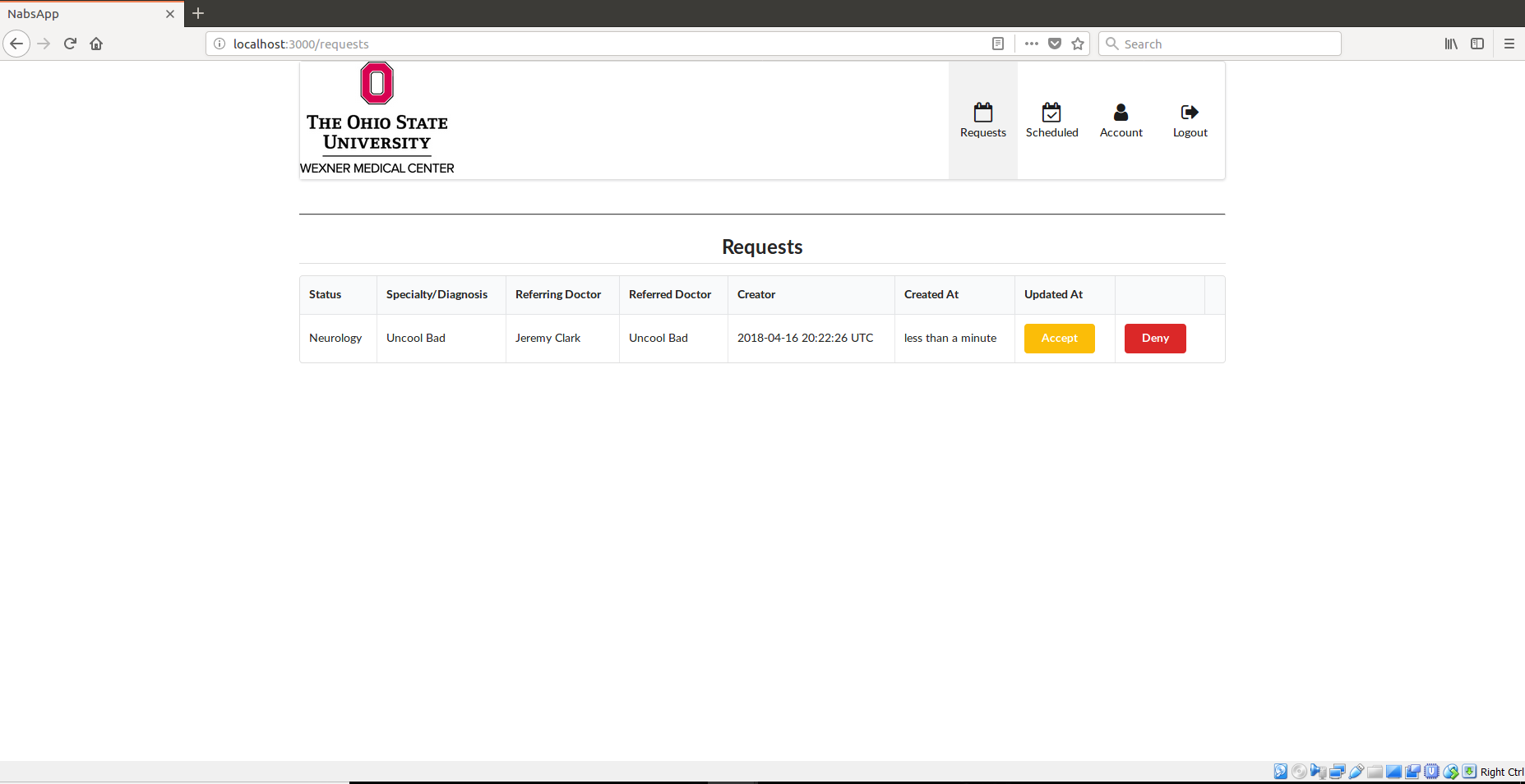


Referring doctors and staff can manage pending referrals that have not yet been sent.

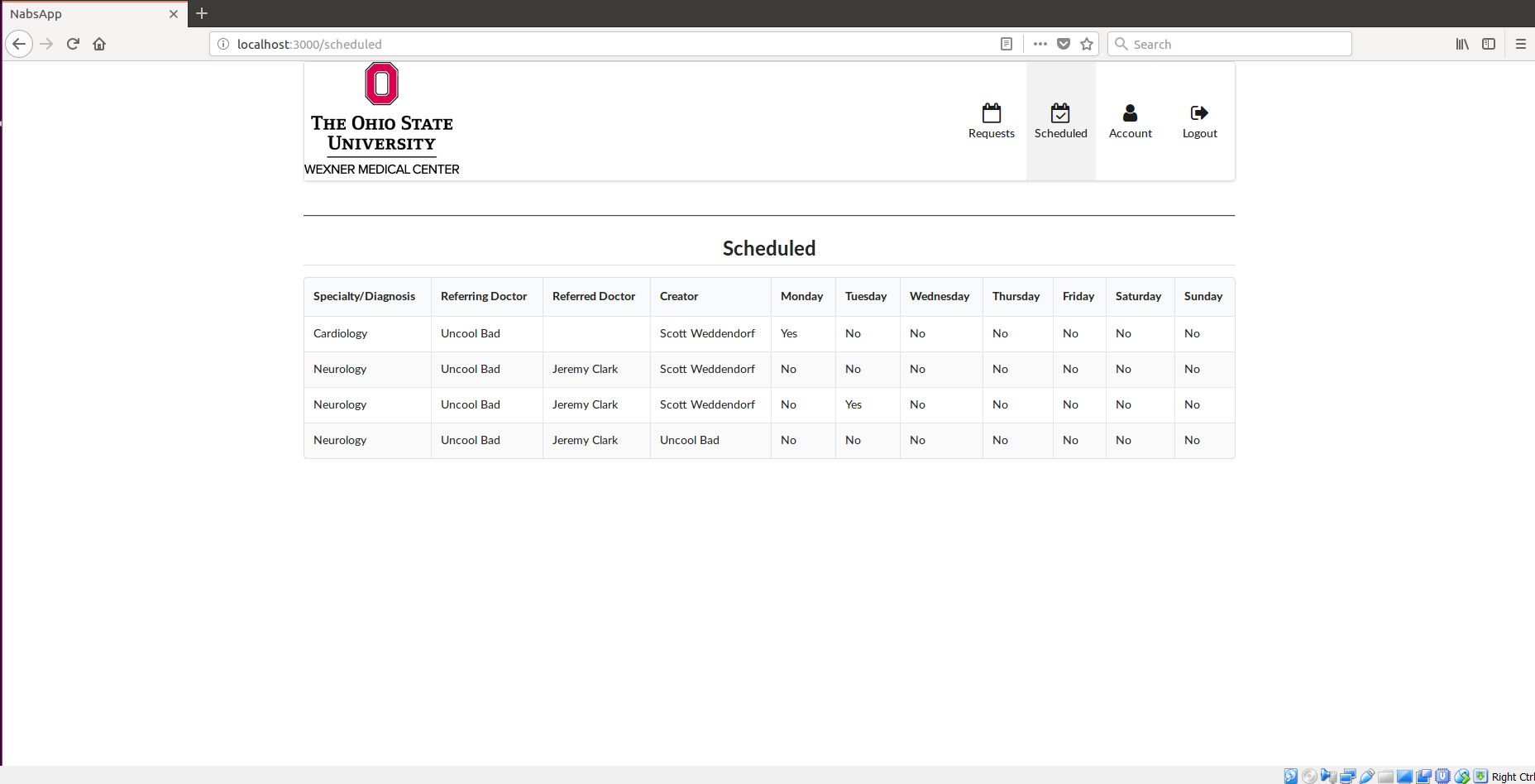
**Referred Doctors and staff:**



Referred doctors and staff are able to manage incoming requests to their clinic, view accepted referrals in the scheduled page, and manage their own accounts.



Referred doctors and staff can accept/decline requests sent to them. A referred doctor also has the option of auto-approving referrals within their “account” page.



Referred doctors and staff can view the accepted referrals for their clinic.

**Deployment Guide**

Deployment was not in the scope of the project since everything is being done locally. In the future, the project will be deployed, but that was not something dealt with this semester.

**Project Plan**

|  |  |  |
| --- | --- | --- |
| **Task** | **Completed?** | **Iteration** |
| Get code from previous team | Completed | 1 |
| Set up project (VM, Ruby, etc.) | Completed | 1 |
| Create Setup Guide | Completed | 2 |
| Create Log In Guide | Completed | 3 |
| Allow clinic email to be edited | Completed | 4 |
| Fix clinic delete error | Completed | 4 |
| Add notice when user is deleted | Completed | 4 |
| Clinic Admin can create referral | Completed | 5 |
| Doctor can approve pending referral | Completed | 5 |
| Make password display correctly on account | Completed | 5 |
| Allow doctor to select specialty when user is created | Completed | 5 |
| Allow for conditions on doctor search page | Completed | 6 |
| Foreign key bug with referrals | Completed | 6 |
| Allow specialization page to have conditions | Completed | 6 |
| Add image header to each form page | Completed | 6 |
| Create buttons on homepage for easy access | Completed | 6 |
| Make denied appointment email go to staff not patient | Completed | 6 |
| Auto-approval | Completed | 6 |
| Phone numbers gives error when not in certain format | Completed | 6 |
| Email confirmation goes to doctor | Not completed | N/A |
| Search function should allow some or none of the filters to be selected | Not completed | N/A |

**Known Issues and Feature Expansion**

The following is an unordered, unprioritized list of known issues and possible feature expansions for the NABS app.

* General resiliency issues. There are possible instability bugs (and there will likely always be some). So, a good practice is to have good error checking and fallbacks. Possible improvements for this would be to have a /error/ route, a global error catcher, or just ensure error checking is done for all error prone code. Without some sort of error checking, a page will completely crash and show stack traces. Which is a security violation, and poor user experience.
* A different doctor within the same referred clinic seems to be able to accept referrals meant for a different doctor. Ideally, A doctor should only see/accept/decline referrals for themselves, or if they accept a referral for another doctor, the referral is still assigned to the original doctor. This does not seem to be the case currently. Clinic staff should be able to view and accept/decline referrals for all doctors in their clinic. This functionality is there.
* Clinic staff should be able to view/manage the users within their own clinic. Basically, they should have a view similar to the admin’s users view that shows only the users within their clinic.
* Add functionality for dynamic drop-down menus wherever they are. Some examples are the doctor search page and the referral creation page. The dropdowns should only display consistent information based on the other dropdowns. So, if you select a certain specialization on the referral creation page, then the referred doctor dropdown should only display doctors that have that specialization.
* Add the ability to upload a file to be emailed with the patient confirmation email. Currently, the email that is sent to the patient is very bare bones, this should be expanded upon with either embedded content or a file attachment. The file would need to be attached after the referral is accepted, before the email is sent to the patient. This would likely require another possible state for referrals since the auto-approve function complicates the usual workflow.
* Add appointments to the domain. Currently, referrals and patients are a part of the domain model but there is no way to enter an actual appointment with a set day and time. This was prevented because we were not able to interface with the hospital’s scheduling system. But, there should still be a way to manually enter an appointment into the system. This would likely relate closely with the file upload issue mentioned earlier.
* The code should be refactored to better follow general best practices. There seems to be a lot of redundancy, complexity, and code duplication. The codebase would benefit from the addition of a service and possibly a DAO layer so abstract some of the more complex and redundant code away from the controller classes. The controllers are meant to be simple and cohesive, but a lot of rapid prototyping and loose communication has turned them into large, complex classes.
* Deployment of the app into a test environment so the sponsor can give more accurate and constant feedback to the team.

**Development Guide and Git Repo**

**Set Up:**

1. Get VirtualBox setup with Ubuntu

<https://www.virtualbox.org/>

<https://www.ubuntu.com/download/desktop>

[Everything else is done inside the Virtual Machine]

1. Install RubyMine

<https://www.jetbrains.com/ruby/download/>

OR sudo snap install rubymine --classic

1. Install Ruby

<https://www.ruby-lang.org/en/downloads/>

OR sudo apt-get install ruby-full

1. Get the project from GitHub
   1. [RubyMine] find “Checkout from Version Control”... “GitHub”... (login to GitHub)...
      1. Note: You might have to clone the repo in Terminal to a directory and open it afterwards with RubyMine
   2. Git Repo URL: <https://github.com/BetaRidley/OSUNeuroScheduler.git>
2. Install PostgreSQL 9.5. When it’s all done, make sure pgAdmin III is somewhere easily accessible e.g. Desktop. (pgAdmin III is just a GUI for Postgres.)
   1. <https://www.postgresql.org/download/>
   2. Change Directory into Downloads
   3. chmod +x postgresql-9.5.10-2-linux-x64.run
   4. sudo ./postgresql-9.5.10-2-linux-x64.run
3. [pgAdmin] make a new User/Password/Database.
4. [RubyMine] in the Terminal, run “bundle install”
   1. For anything that you do in the Terminal, make sure that you cd into the ‘nabs-app’ directory (or have the project structure default to that being the ‘root).
      1. If an error occurs with installing the ‘pg’ gem, try running:

sudo apt-get install libpq-dev

* + 1. You do not need the ‘sqlite’ gem to be installed. If errors occur with that gem, comment it out of the Gemfile.

1. Rails and Postgres need to be able to communicate. To confirm this, make sure that the following values are identical:

[pgAdmin] [RubyMine: /config/database.yml]

<The name of the DB> <Any value where key is “database”>

<Your username / Role> <Any value where key is “username”>

<Your password for said Role> <Any value where key is “password”>

1. [RubyMine] In the Terminal, run “rake db:create”, then

“rake db:migrate RAILS\_ENV=development”.

After a few minutes, the Terminal should produce many different SQL-ish statements. Assuming that no errors come up, it means the migration was successful. You can check this by looking at your Database in pgAdmin.

1. [RubyMine] in Terminal, run “rails server”. The Terminal should read something like “Listening on 0.0.0.3000” (the exact number might be slightly different).
2. [Browser] Type in “localhost:3000” in the address bar (or, if the number was different in the previous step, replace 3000 with whatever number the Terminal displayed). You should be directed to the Log In page.

**Log in:**

1. To Seed the Database with an Admin account, put the following line into the file ‘db/seeds.rb’:

Clinic.create(name: ‘Clinic 1‘, role: ‘referrer’)

Clinic.create(name: ‘Clinic 2‘, role: ‘referred’)

User.create!(email: 'youremail@gmail.com', password: '1234', password\_confirmation: ‘1234’, password\_changed: true, created\_at: DateTime.now, updated\_at: DateTime.now, role: ‘admin’)

This creates the Site Admin account, who oversees all User Accounts. The Admin can

create new users of the different roles and clinics, as well as other information. These

values can also be seeded in a similar manner as above. See the comments in the

‘seeds.rb’ file for details.

Note: Replace the email and both password values with a real email and password

combination.

1. [RubyMine] In the Terminal, run ‘rake db:seed’. The Database should now include this admin account
   * 1. To check if it worked, [pgAdmin] right click on the Users table and go to ‘View Data’. You should see your entry.

**Run the App:**

Currently, the app can only be run locally.

[RubyMine] in Terminal, run “rails server”. The Terminal should read something like “Listening on 0.0.0.3000” (the exact number might be slightly different).

[Browser] Type in “localhost:3000” in the address bar (or, if the number was different in the previous step, replace 3000 with whatever number the Terminal displayed). You should be directed to the Log In page.

**Git Repo:**

Git Repo URL: <https://github.com/BetaRidley/OSUNeuroScheduler.git>