## Project Concept

This is a website that will allow UNC computer science graduate students to view their status and progress in the pursuit of their degrees. It will allow them to view all of the courses they have taken as a graduate student, all of the documents they have submitted as part of the program, all of the personal information they have provided to the department, and their current progress on the way to their degree. They will be able to see what parts of their degree requirements they have fulfilled and what else they have to do. This website will allow them to view only their own information and will be gated behind an onyen login.

This website will also provide a way for our client, Jodie Gregoritsch, to view all of the students' information and also a way to edit any of this information. She will be able to view any students' profile and be able to leave notes for the students on their profiles about their current status and progress. It will provide a place for her to post the grades of the students in a centralized location so that they can find their own grades without having to go through her. It will also provide a visually pleasing and efficient way for her to view the information of any student.

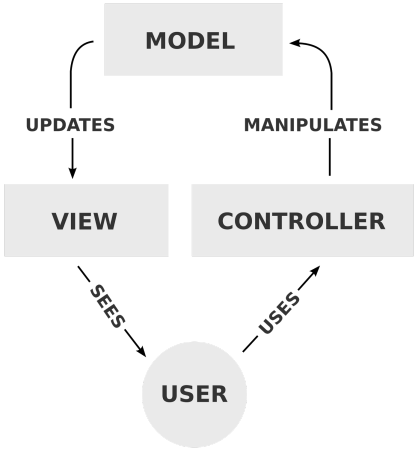
## Tweet

Grad Tracker: a website for graduate students to be able to find their progress and grades through school all in one place

# Design Document:

### Architecture diagram:

Our system follows the basic MVC architecture system. Sample diagram as follows:




https://upload.wikimedia.org/wikipedia/commons/thumb/a/a0/MVC-Process.svg/2000px-MVC-Process.svg.png

**Architecture**

- View
  - HTML Page [It's a single file]
  - Stylesheet and JavaScript
- Model
  - Students Table
  - Course Histories Table
  - Course Descriptions Table
  - Documents Table
  - Jobs Table
- Controller
  - Make the queries to the table. Allows global variables [student data, basically] to be accessed by view.

**Decomposition**

- Modules
  - Controllers
    - Used to talk to the database and control the flow of information to the views
  - Migration Files [Used for creating the tables]
    - Create Students Table
    - Create Course Histories Table
    - Create Course Descriptions Table
    - Create Documents Table
    - Create Jobs Table
  - Views
    - The files that are actually seen by the user. Mainly html and js files that are run client side. Talk to the appropriate controllers to get information on what to do and to get information from the database

**Design Decisions**

- Implementation using Ruby on Rails.
- Database only contain the current students (security concern), no inactive student/data will be available.
- Easy access of data  using MVC architecture.

# Administrator Manual

- First, install Ruby on Rails. We are running version 2.3.1. This can be online at rubyonrails.org
  - Then, install the MySql gems and appropriate JavaScript gems
  - Then, import the appropriate github repository of the site
- Run the appropriate Rails commands to set up the database and run the migrations
  - Db:migrate is used to run the migration files to create the tables
  - Db:seed is used to seed the database with the default values
- How to create users and user roles
  - Admin will have the ability to add new students to the system from the add student tab in search page of the website
  - New faculties and additional admins need to be added directly to database.

# User Manual

User manual

- Get started:
  - In order to get started, you will need internet access and a web browser.
  - Access the login page at findgrade.me
  - To access, please log in to the system with your pid
  - Faculty
    - On login, you will see a search page
    - Search for the student/s you are looking for in the text boxes and click on the square that comes up with his/her name to go to his/her student page
    - Use the tabs on the side on the left and at the top on the right to navigate around the page
  - Student
    - On login, you will see your personal information page
    - This page will contain all of the information the site contains about you
    - Use the tabs on the side on the left and at the top on the right to navigate around the page
    - Use the files tab to submit files to the department
  - Administrator
    - On login, you will be brought to a search page

        - Use the buttons labeled ‘Add Student’ and ‘Delete Student’ at the bottom of the page on the left to add and delete students respectively
        - Use the fields and text boxes on the left to search for the student/s you are interested in
        - Click on a box labelled with a student name on the right to go to their student informational page
        - Use the tabs on the side on the left and at the top on the right to navigate across the page
        - Click on any field contained on the page to edit the information contained in it and then click submit at the bottom of the page to change that information in the database
- How to register
    - Email the department asking for a UNC Computer Science email and id
- Specifics about fields
    - The login field on the login page should be your UNC pid
    - The search field on the search page should contain only alphabetic characters. The search will find all students with names that contain the word entered in the field
    - The “Add Student” button should only be pressed when you have a document that contains all the fields needed for the add function and those fields contain the right type of data, ex. a number in a field that’s for numbers
- Troubleshooting
    - If you have any issues logging in, email the UNC computer science department help desk to make sure that your CS id is still valid
    - If you are a student and cannot view your own page, logout and log back in. If that doesn’t work, contact the administrator of the site directly
    - If you are a faculty member and cannot view everyone’s page, logout and log back in. If that doesn’t work, contact the administrator of the site to make sure you are in the faculty table.
    - If you are an administrator and cannot edit everyone’s information or cannot view everyone’s information, logout and log back in. If that doesn’t work, check the database of the site to make sure you are in the administrator table and have the requisite permissions.

# Test Plan :

**Ideal Plan**

- *Common Test Cases*
    - Everyone is able to log into their correct account.
    - Check that our website is browser compatible. [i.e. Firefox, Chrome, Safari, and Internet Explorer]
- *Admin*
    - Can import courses taught by the department.

    - Can import batches of students and their personal information through csv or something similar.
    - Can add a single student into the database.
    - Can import student's' courses. [i.e. Students in a particular course. ]
    - Can import documents associated with the student.
    - Can delete documents.
    - Can edit all information associated with the student.
    - Submit updates the database properly.
    - Can delete a student or set a student inactive.
    - Can view notifications of students. [i.e. Pending Documents]
    - Can approve or reject pending items.
    - Can give a explanatory note to student for rejected items.
    - Can search for students by milestones or by any other field in the database. [i.e. Citizenship, Residency, Funding Status, etc]
- *Faculty*
    - Can search students by name, PID, or email.
    - Has the same searching capability as the admin.
    - Can only view all of the students in the database.
    - Students information, course history, documents, and requirements are not editable.
- *Student*
    - *Can only view their account.*
    - *Can upload documents (which will notify the admin).*
    - *Can see the current requirements met in their program.*
- *Freelance Testing*
    - *Have a working website interface, correct functionality, and appropriate styling.*
    -

**Intended Plan**

Since our program is small enough, we believe it is possible to test all cases set forth in our ideal plan.

**Functional Spec**

**Use Cases**

*Administrator Role*

1. To log in.
    a. From the homepage, the admin is redirected to the onyen authenticator where he/she will enter their onyen and password.

i. Log in successful: redirected to the search page of the website.

ii. Log in unsuccessful: stay on the onyen authenticator website until a correct onyen and password are entered.

2. To find an individual student by name or PID.

    a. Admin enters the name or PID in search box located on the side bar of the website.

    b. Results displayed on right side of the same page.

3. To get a list of students meeting or not meeting criteria Y.

    a. Admin will click on advance search located in the side bar.

    b. Specific filter opens will appear.

    c. Admin will enter criteria and click search to retrieve the results.

4. To add a student.

    a. Click on add student

    b. Admin will be directed to a form to complete

    c. Click submit.

        i. If input is missing or formatted incorrectly, pop up will alert the admin that there is an error with the input.

        ii. If input is correct, admin will be redirected to the new student's page.

5. To imports a list of students

    a. Click on import students.

    b. A file selector will pop up and the admin imports excel or csv file.

i. If a user exists in the database already and is also being imported

1. Alert admin whether to keep original data, update user data, or to replace student data.

ii. Correct File Formatting:

1. Database is updated will the list of students.

iii. Incorrect File Formatting:

1. Pop up will alert the admin that the formatting is incorrect. Not changes will be applied the database.

6. To mark a requirement complete or incomplete for a student.

a. On the student's page, click requirements

b. Modify any checkbox or date.

c. Click submit.

1. To make edits to a student's personal information.

1. On the student's page, click on personal information.

1. Edit the appropriate fields.

1. Click submit.

1. To make notes.

1. Where shown [beside individual student data], click 'add note'.

1. Write note and select if a student can view it too.

1. Click done and admin [or student] can now view note later on.

1. To upload or view a document for an individual student.

1. On the student page, click documents.

1. One the document, select the file and description.

1. Click submit and the file is uploaded for that student.

1. To export student's data as a PDF.

    1. On the student page, select personal info, requirements, or courses.

    1. At the top right, click export.

    1. File will be downloaded.

1. To view documents pending approval.

    1. From the admin's search page [home page as well], the admin can click on notifications.

    1. Admin will be redirected to the page containing all of the pending documents besides the student's name.

13. To approve requests

    1. Admin will have option of two buttons besides each item pending approval. Will select either approve button or deny button.

1. To change/upload grades
    1. Change a grade
        1. Admin will navigate to a students' page from the search page and select course history tab.
            1. Admin can manually edit or add any grade to any class on the page.
    1. Upload multiple grades
        i. Admin will have an excel or csv form and upload it to the site. Site will parse the information and add the correct grades to the correct students.

*Faculty*

1. To log in---same as admin

1. To view student information

    1. Faculty have accessibility for student search page, and the process would be same as admin

    1. Once the faculty gets into a specific student's page, the faculty would have the same rights as student (not being able to edit anything)

*Student*

1) Student views information concerning grades

    a. User clicks on the courses tab on left which will bring up Course History tab.
        i. Tab will have all of the classes they have taken as a graduate student. Classes will be organized vertically by semester/year with most recent/current at the top.

2) Student views information regarding submitted documents

    1. User clicks the documents tab on left to open Documents tab.

        i. Tab will have a section called submitted documents which has all submitted documents by user.
        1. They will be organized vertically by date with most recent at the top.

3) Student wants to add a document

    a. User clicks the documents tab on left to open a Documents tab.
        i. Tab will have a section called Add new document.
            1. User will choose what file to add, what type of file it is, and then click add file to add the file to their account.

4) Student wants to check on their personal information
    a. User will click on personal information tab on left.
        i. User can look at what data the department has for them.

5) Student wants to check on their degree requirements
    a. User will click on degree requirements tab on left.
        i. They can select either Ph.D. or Master's tab
            1. Each tab has collapsible tabs that contain information on what requirements for the degree they chose above they have satisfied.

2. Information and requirements will may be repeated across Ph.D. and Master's tab.

**User Stories**

Student

As a student in the middle of graduate school, I want to be able to easily see in one place where all the forms I've submitted to the department and my advisor are. I want to be able to see what I have already submitted to the department so that I can stay organized with what I need to apply for things such as graduation.

Student

As a graduate student, I want to be able what classes I've already taken and what classes I am taking right now in one place. I want them to be organized neatly so that I can see when I've taken them and what my grade was/is in them.

Student

As a student, I want to be able to submit needed forms to the department. I want to have one central location where I can submit all my needed forms and I want the process to be very easy and streamlined.

Student

As a student, I want to be able to see my progress to getting my degree in one place. I want to be able to see what requirements I have satisfied and with what so that I can plan what classes I need to take.

Administrator

As the main administrator, I oversee the graduate student tracking and manage the data. I need a program that can allow me to approve and finalize any student information that they've sent in.

Administrator

As the person in charge of graduate studies, I want to be able to check on the progress of all the students in the program. I want to be able to see who has satisfied which requirements and I want to be able to check off what requirements they have satisfied in a way that is easily visible to them.

Administrator

As the main administrator, I manage the grades of the graduate students. I would like to be able to easily edit student grades and also upload many grades at the same time.

Faculty

As a faculty in the CS department, I want to be able to see students' information and their progress towards their degree. If there is a problem, however, I will need to notify the administrator to make changes.

**Requirements**

Priority code: 1 is most important, 4 is least important

Database

(1) Create administrator and student identities

(1) Create schema to store all user's personal information

(1) Create schema to store all students' grades

(3) Create schema to store students' files

(3) Create faculty user type

(4) Create schema to store user photos of students.

Search page

(1) Create way to search for students by name or PID

(3) Advanced search- search database for certain requirements and create lists of students who meet those requirements

(3) Expanded advanced search- create groupings of requirements called milestones that can be searched by. Students who meet the selected milestones while not being part of a milestone further ahead become part of the list.

(4) Mailbox- alert function that displays a message that a certain student has done something that requires administrator approval. Flagged on a 'pending' click by a student

(4) Allow faculty user type to upload grades through excel or csv file reading

(2) Allow administrator to add/delete individual students from the database

(3) Allow administrator to add batches of students to the database through excel or csv file reading

(2) Allow administrator to add many grades at one time on an excel or csv file. The system will then associate those grades and classes with the appropriate student in database.

Login page

(1) Tie website to UNC onyen api so that onyen authentication is required to enter site

Information page

(1) Page that has the information of certain student on it- must be tied into database and bring up their personal information, grades, and progress towards degree. This will create three tabs on the left, one for each above category.

(1) Must create two inner tabs in the 'progress towards degree' tab- one for Master's requirements and one for Ph.D. requirements

(1) **Constraint** Must have two different views with the same information. Administrator view and student view. Must be flagged what type of page to create by what type of user they are in database. Admin view must be entirely editable, student view must be not editable at all. Faculty view must not be able to edit anything, but can view any student's information

(1) Must fill out information on the page according to the stored information in the database for that user- fill out all of the personal information, course history, degree requirements

(3) Create file tab for students.

(2) Make degree requirements tab dynamic. Make it show the students' own progress/completion of each requirement.

(3) Allow students to upload files to their account.

(3) Allow students to view the past files they have uploaded.

(4) Create a pending system for files and other requirements on other tabs that alerts the administrator they believe they have done something. Administrator will then either approve or deny.

(4) Create place on page to allow student to upload and display a photo of herself

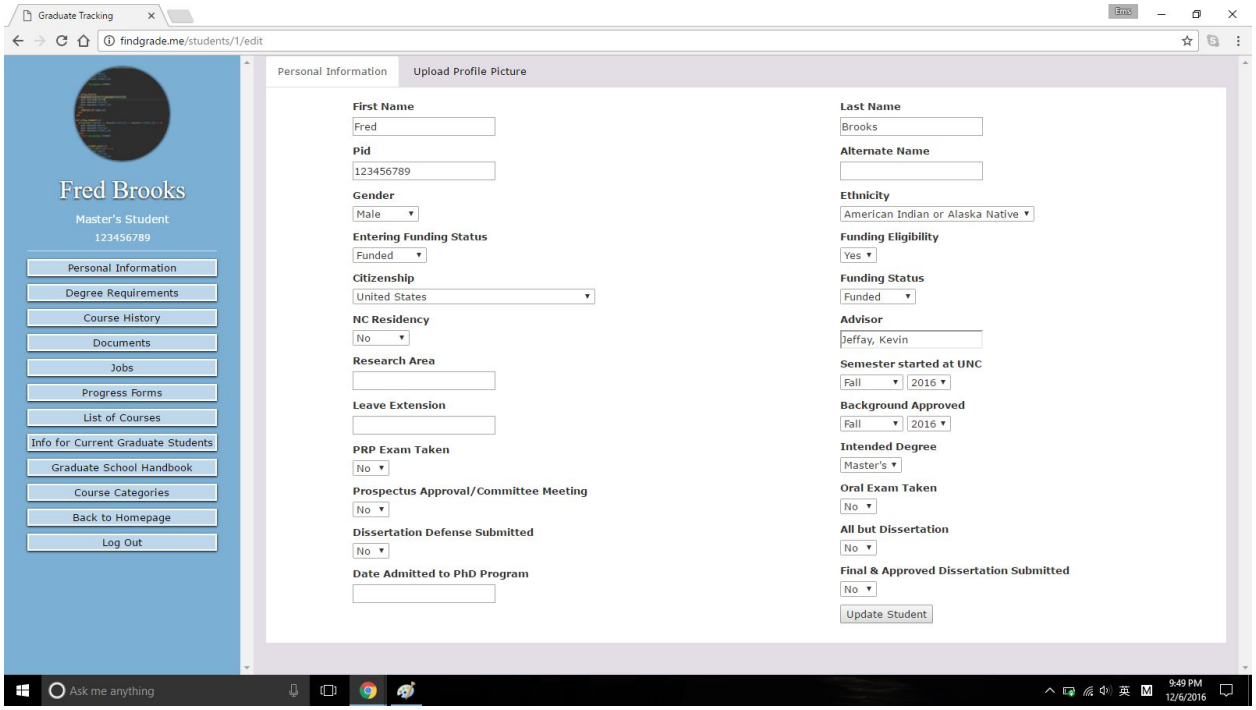
(4) Use existing databse on cs.unc.edu of graduate student photos to populate the photo display by default

## Interface

Student Informational Page

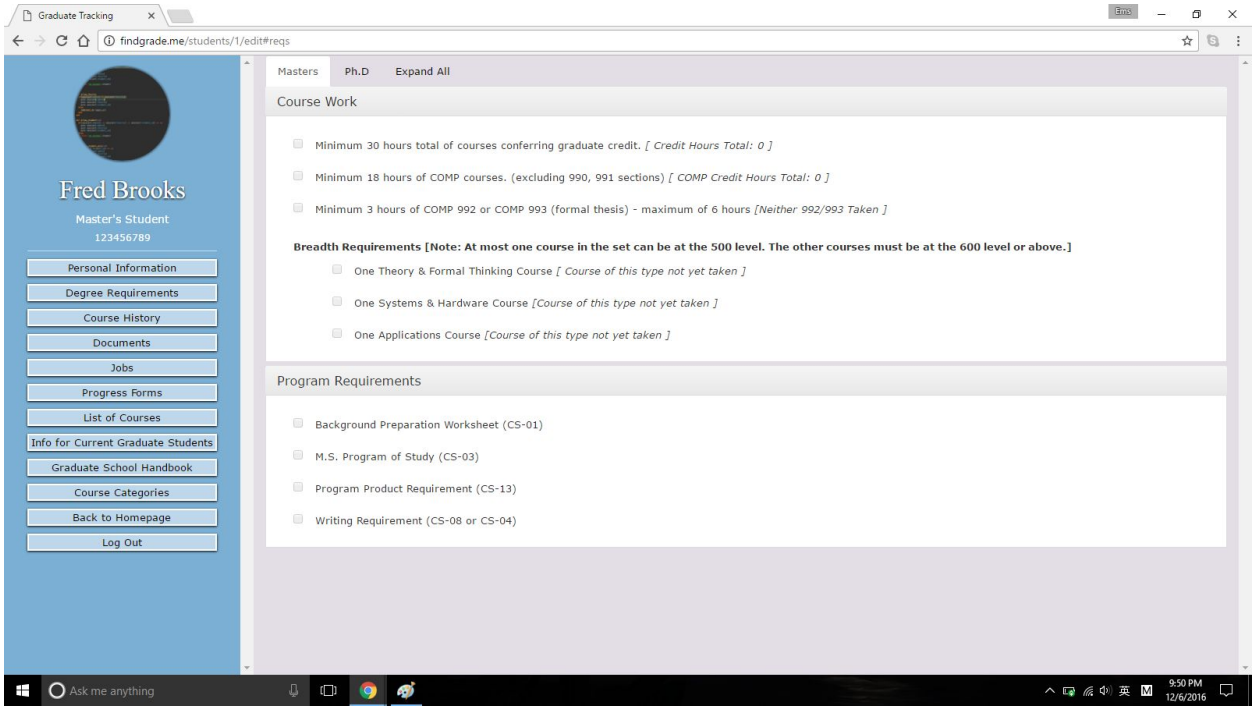
- This is the page that all information about students will be shown on.
- Each tab on the left will bring up information related to its name on the right.

Personal Information



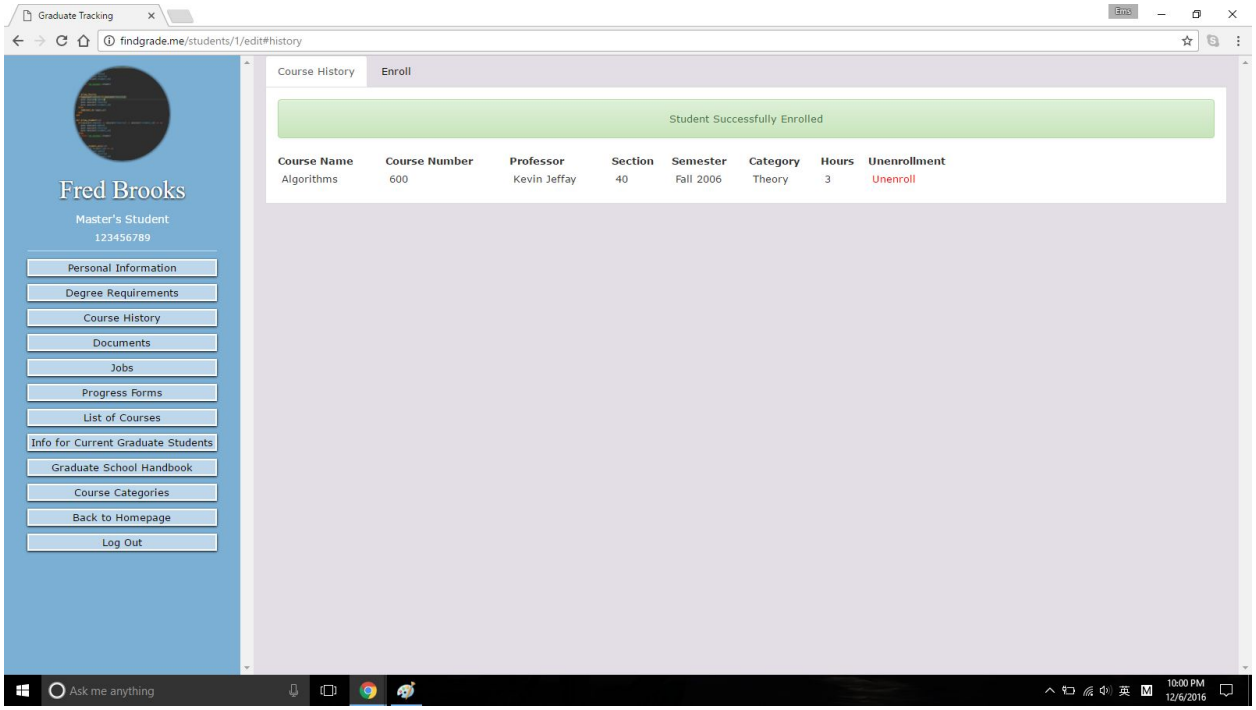

- Contains all the personal identifying information provided to UNC and the CS department by the student

Degree Requirements



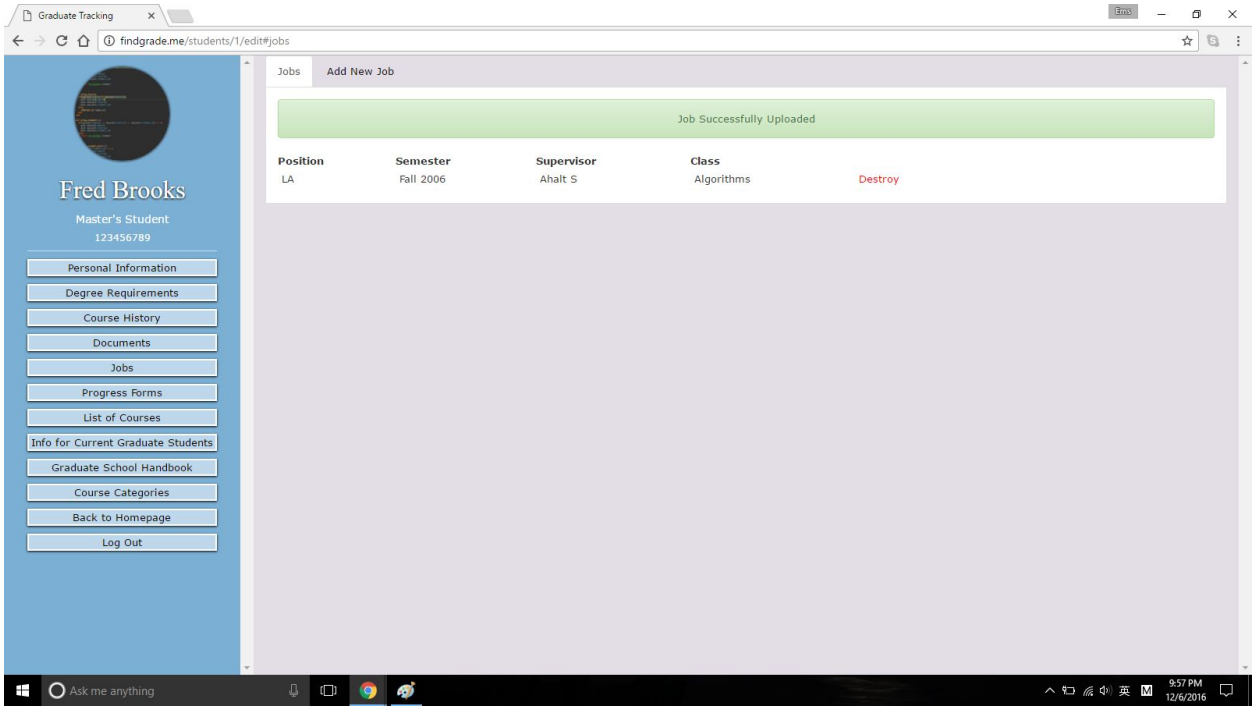

- Right side of the screen contains three tabs at the top: Ph.D., Master's, Expand All
- Ph.D. contains all the progress the student has made towards the Master's degree
    - 3 accordions separate the data up
- Master's contains all the progress the student has made towards the Master's degree
    - 2 accordions separate the data up
- Expand all expands all the accordions on the current open tab

Course History



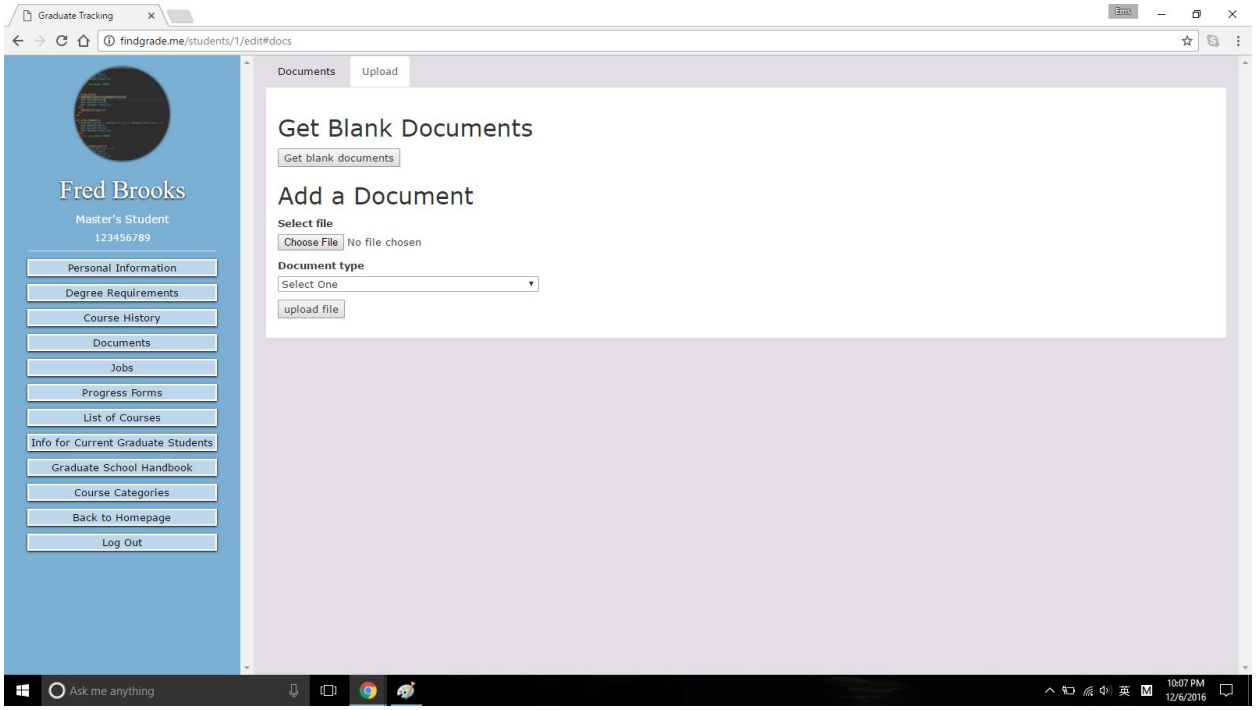

- It displays a table containing information on all the courses the student has taken at UNC as a CS graduate student.

Jobs




- Contains history about all the TA/RA jobs the student has held while being a CS graduate student.
- Separate add tab to add new jobs.

Documents




- Students will be able see their past uploaded files on the documents tab
- On the upload tab, students will be able to get blank documents from department site, and add filled-in forms to their documents page.

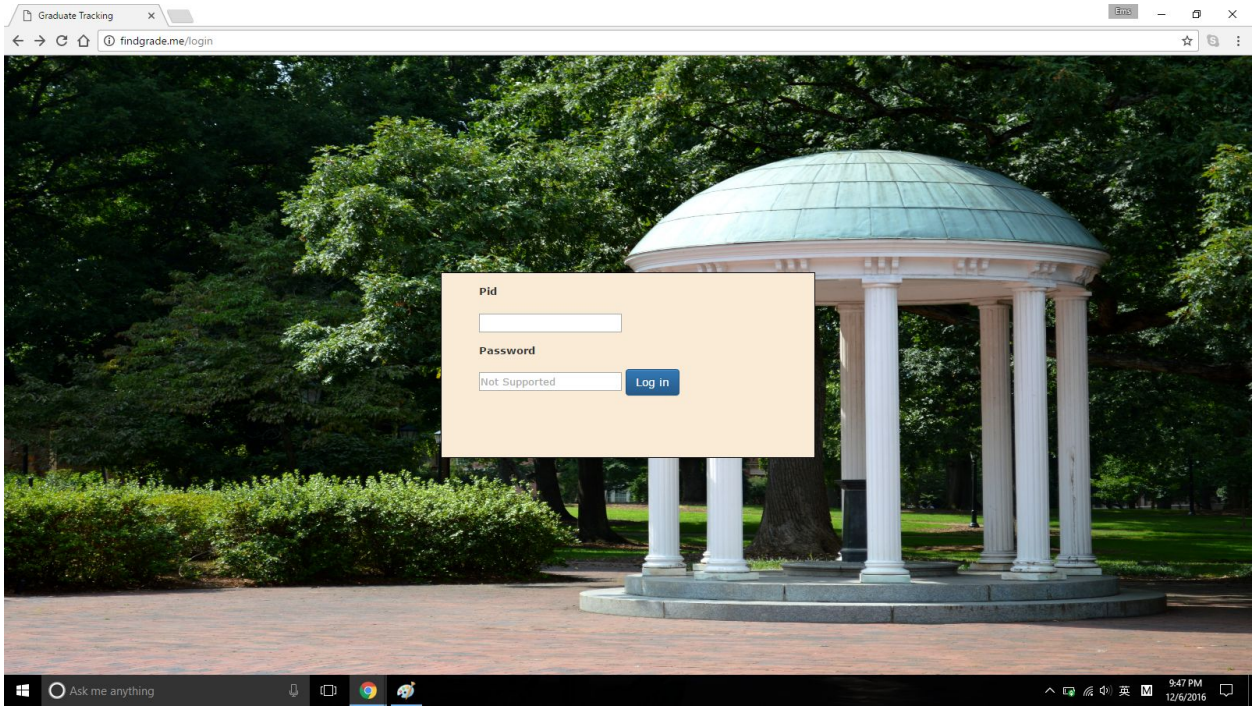
Student/Faculty View

- The information and tabs on the left will be the same as above
- All of the information contained on the right side of the screen will be non editable
- Information will be the same as above
- All of the information will be in labels, radio buttons, etc

Administration View

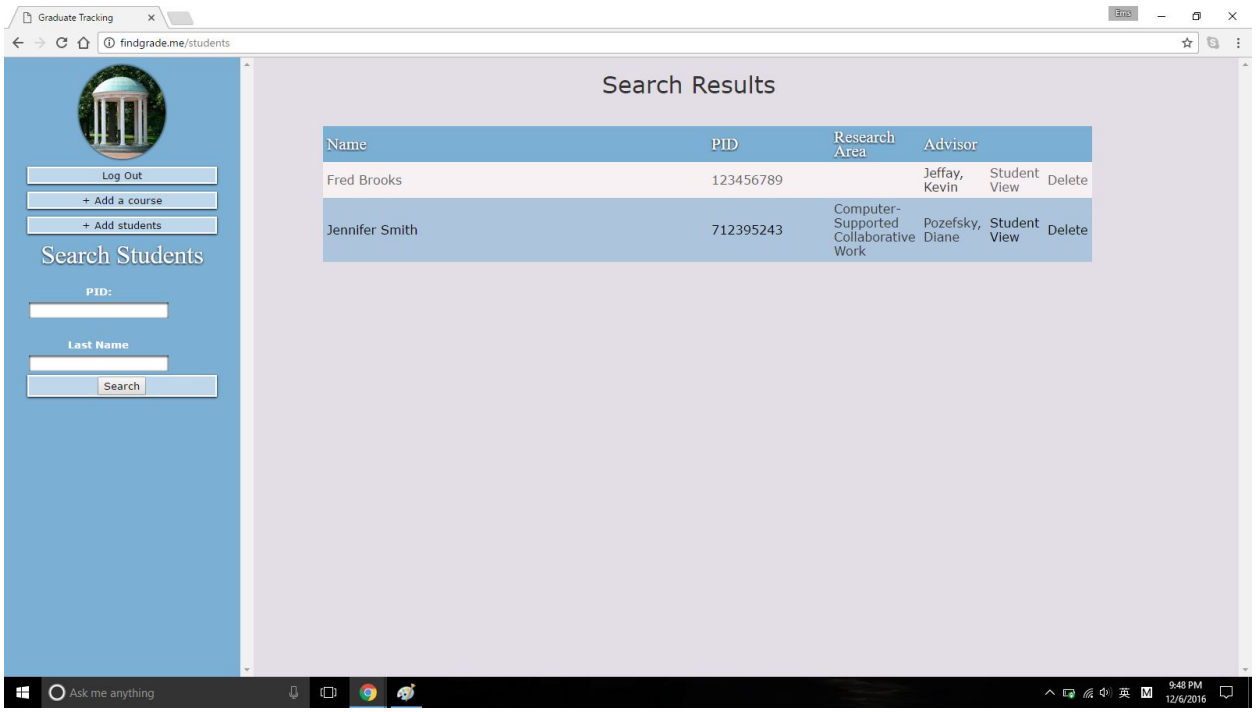
- The information and tabs on the left will be the same as above
- All of the information on the right side of the screen will be editable
- All of the information will be the same as above
- Everything will be contained in text boxes, radio buttons, etc for easy editing

Login Page




- Login page uses pid and password to authenticate user. Current system only utilizes pid field.

Search Page




- Two major parts to this page: left side and right side. Right side will be around 66% to 75% of the page
- Left side contains search fields where administrator or faculty can enter search criteria
    - Current criteria  are pid and last names.
- Left side also contain buttons for logout, and adding students & courses to database.
- Right side contains boxes stacked vertically with the names of the students that matched the search parameters
    - Clicking on a box will link your page to the Student Informational Page of that particular student

**Personas**

Lucy (28)

New Hire as an Administrator.

Lucy just graduated with her associates degree and started her position as an administrator for the department. She's done some similar work before when she interned for her state senator. Other than that, she was a sales associate for 3 years. About her personal life, she is married and has one child. Lucy lives about 30 minutes from her workplace. Time is very important to her

and she likes to be efficient, tidy, and clean. She wants to be able to manage the graduate students she is in charge of in an efficient manner.

Henry (26)

*M.S. Student*

Tim is in his last year of his program and is spending the majority of his time working on his research which is mainly data mining. He is also looking for a position in England, his home country, after his graduation. His field of expertise is data science, but he also has a B.A. in Economics. He isn't sure if he wants to pursue a Ph.D. yet. He wants to be able to see what requirements of a Ph.D. he has satisfied already while getting his Master's degree.

Joe Smith

Faculty

Dr. Joe Smith is part of the faculty working in the CS department. After getting his Ph.D degree, he moved down to North Carolina, and now is happily married with two kids. At work, his main interest is in graphics, and Dr. Smith is both responsible for teaching a graduate level course and serving as an advisor for several graduate students. With these responsibilities, he often communicates with the administrator to update on student grades and make other similar inquiries.

Harry Potter (23)

Ph.D. Student

Harry Potter is a first semester PH.D. student. He is studious and is competent with computers. He likes being able to see his all his grades and requirements in an easily accessible location and likes being to check up on his current standing in his program whenever he wants. His goal is to go to his classes every semester and graduate on time with a PH.D., with the option of picking up a Master's degree on the way. He dislikes inefficiency and likes being able to take care of things himself. He is a North Carolina native and has always planned on graduating from Chapel Hill with a PH.D. in Computer Science. He lives off campus and is beginning to adjust to both the university's social scene as well as its academic workload.