GetInSPIREd

Final Document

Team Goose

https://youtu.be/1bgpnTplcgE

Development Team

Tim Contois worked mainly with the database, the major page, and implementing the getInSPIREd button. On the user page, he made it possible for users to select a major and concentration and then an algorithm checked what requirements were satisfied. On the getInSPIREd page, he worked on 3 algorithms to help users pick majors. One tells a user the closest concentration to completion given a major. Another tells the user the closest major and concentration to completion out of all majors. The last one tells users majors requiring a certain course. He also helped with the UI.

Alex Lassalle designed the mockups and did a lot of work with the .handlebar files, he created the header and the footer for each page as well as the design of the team page. He also worked with the rest of the group adding major requirements to the database. In addition to the page's design he helped organize the features so that they auto adjust to pages changing width.

Josh Pikovsky designed the views and routes for the admin pages, and the initial framework for several other views including courses and course details. He also provided much of the writing in the proposal, readme, and final document.

.

Andrew Gramigna worked mostly with the user's profile page by allowing users to change their password or delete their account. He made sure a user could not update their password without having proper credentials. He also designed the logo for the project. He also helped with writing the scripts for the screencaptures.

Riley Condon worked with the About page handlebars in order to help design the overall layout of the page. He also dealt with fetching all UMass professor ratings from ratemyprofessor.com in order to include them in a professor search feature. With this information he was able to create the Professor Ratings page that lets a user input a professor's name and display the appropriate rating information.

Niketh Murali was integral in the construction of the databases that the site relies on. He contributed much of the code to db.js, and was responsible for structuring the system of prerequisites. While his primary work was in databases, he also helped out with a variety of tasks throughout development.

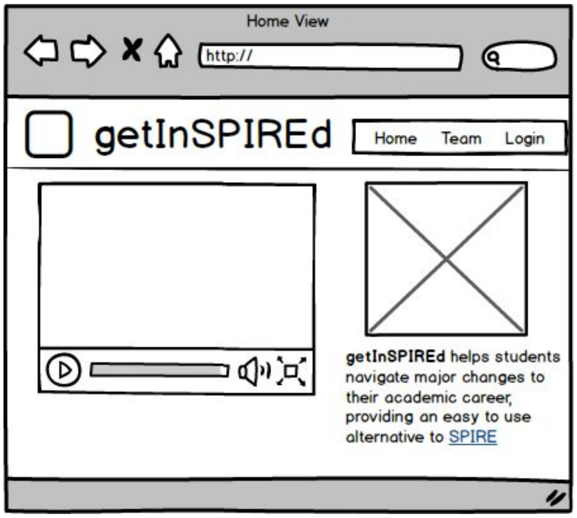
Overview

Our application is a web-based solution to the problems created by Spire, UMass' proprietary academic system. While Spire is a functional application that has some useful capabilities, it is pretty unhelpful when it comes to choosing classes and making academic decisions. Due to some unintuitive design, a user is unable to view their academic requirements as they are choosing classes, or even view their shopping cart while enrolling. Our application focuses its scope specifically on the process of selecting classes, and attempts to create a clean and helpful way for students to see their academic progress. By inputting their major and the classes they have taken, a student will have the option to see the concentrations they are closest to completing, and if there are more promising academic options. Instead of having to navigate through several different sources of information in search of an efficient academic path, getInspired provides a student with all the details they need to make an informed decision about their classes.

Front-End Design

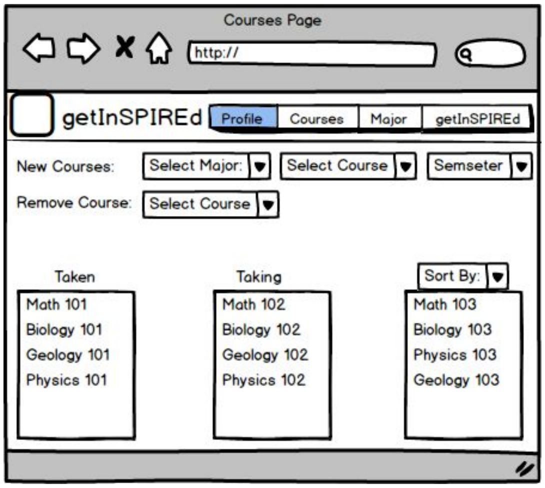
GetInSpired uses a minimalistic approach to the front-end design, focusing mostly on easy navigation and intuitive menus to help guide users to the information they need. By using CSS and HTML, with JavaScripts for dynamically generated content, our pages are clean, easy-to-use, and display useful user-specific information. In terms of aesthetics, we decided on the blue-yellow color scheme that is consistent throughout the site, and then turned our focus more towards usability and ease of use.

The first important view is the home page, which greets the user and provides a brief blurb describing the purpose of the app, along with links to the About, Team and Login pages. This page is kept simple, as it is meant to introduce the user to the concept of the application, and to direct them to either more information, or a way to log in and start checking out their options. As with the rest of the site, the header and footer are kept consistent, and the format of the page is preserved through resizing.

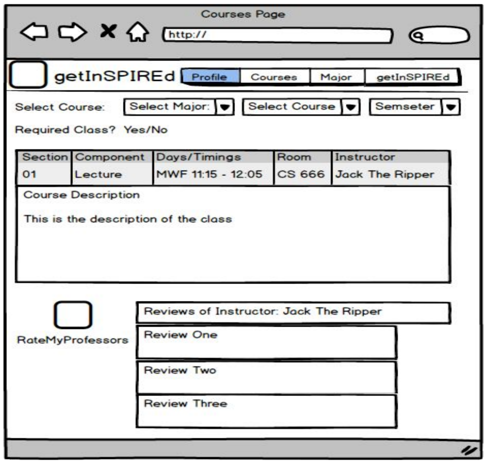


getInSPIREd helps students navigate major changes to their academic career, providing an easy to use alternative to SPIRE

The next important view presented after login is the Profile page. This view provides a user with a drop down menu to select their major, a form to change their password, and a button to delete their getInSpired account. After logging in, several menu buttons become available at the top of the page. This new header provides the user with links to several other helpful pages, and remains for every page while the user is logged in. Another important aspect of this view is the dynamically generated list of classes that the user has added to their profile, and the total major credit value shown on the right side of the page. This profile page gives the user several useful options, and serves as the jumping off point for the rest of the application.



After looking at their profile, the user will most likely proceed to the courses view. This page initially contains a dropdown menu for enter the course department and a dynamically generated list of classes associated with the user. Upon entering the department, and additional drop-down menu appears to prompt the user to enter the course number. Upon entering the number, the course will appear in the list. The design of this page is simple, and it allows the user to effectively view and add to their list of courses. As with the profile page, the user is presented with login-specific menu items at the top of the page.

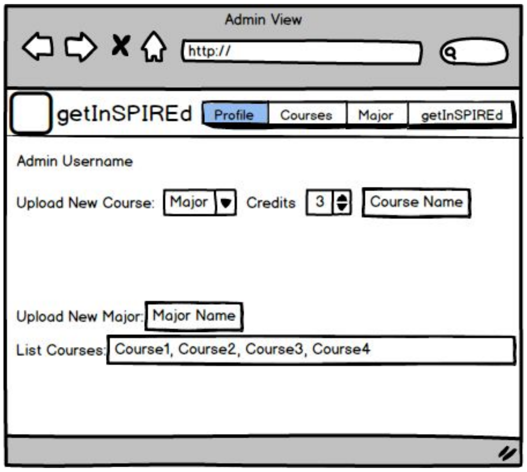


Section	Component	Days/Timings	Room	Instructor
01	Lecture	MWF 11:15 - 12:05	CS 666	Jack The Ripper

Course Description

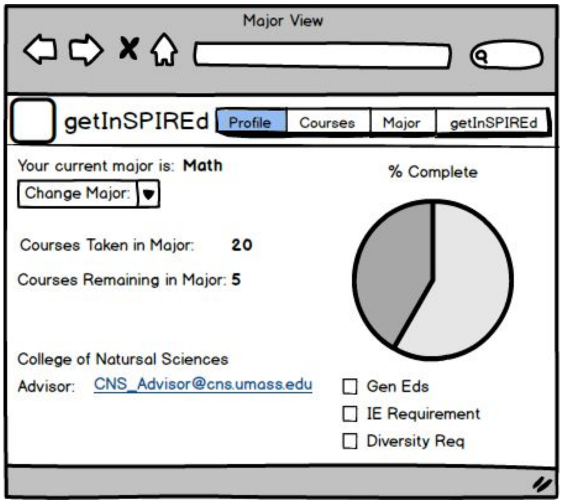
This is the description of the class

The Major page contains much of the proposed functionality of the site. Upon visiting the page, the user is given the option to have the site suggest a major based on their completed requirements. The “Get InSPIREd” button takes the user to a new view, which has several drop-down menus that help find the concentration they are closest to completing.



The Admin page, which is only accessible by users that have been granted admin privileges, provides options to alter users, courses and requirements. Upon clicking the corresponding button, the admin will be directed to a page with dynamically generated content related to the specific view they are in. For example, if an admin clicks the button to add users, they are taken to a page with a list of all current users, and a form to add a new user and set their admin privileges.

Another important view is the Professor Ratings view, which has a form that takes in a professor’s name, and returns some basic information about them pulled from RateMyProfessor.com.


Major View

Your current major is: **Math**

Change Major:

Courses Taken in Major: **20**

Courses Remaining in Major: **5**

College of Natursal Sciences

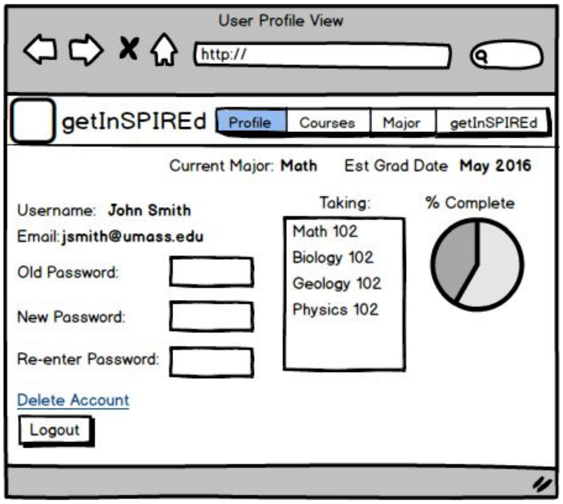
Advisor: CNS_Advisor@cns.umass.edu

% Complete

□ Gen Eds

□ IE Requirement

□ Diversity Req


User Profile View

Current Major: **Math** Est Grad Date **May 2016**

Username: **John Smith**

Email: **jsmith@umass.edu**

Old Password:

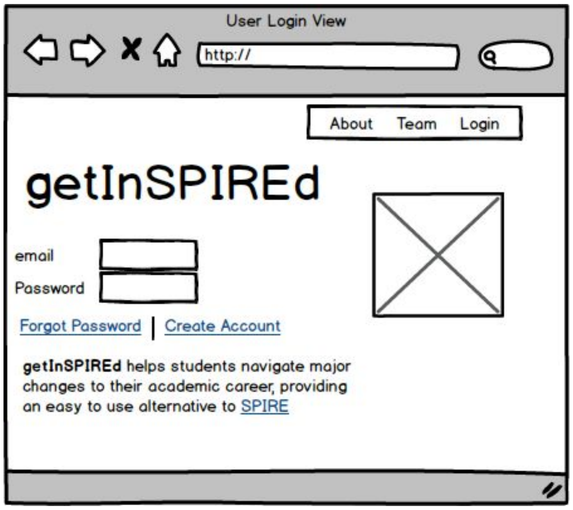
New Password:

Re-enter Password:

Taking:

Math 102
Biology 102
Geology 102
Physics 102

% Complete


User Login View

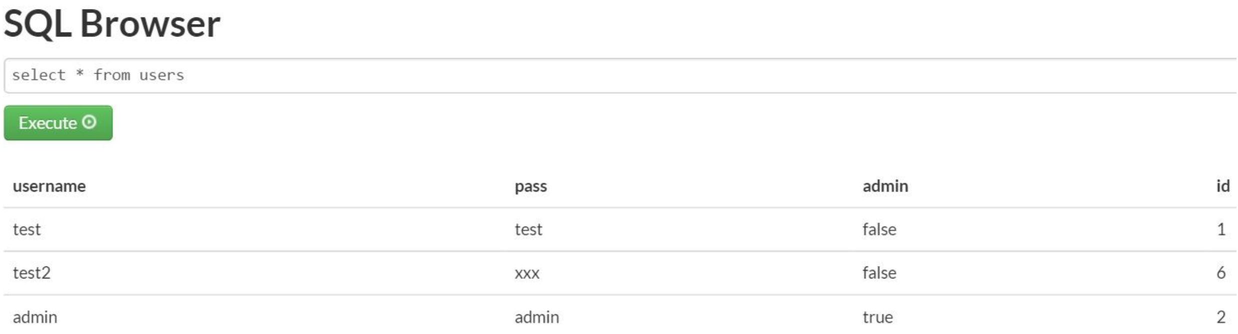
email

Password

getInSPIREd helps students navigate major changes to their academic career, providing an easy to use alternative to SPIRE

Back-End Design

Our application uses an Elephant SQL Postgres database to store information about the logged users, courses, academic requirements, and mappings between users and their inputted courses. This is an example of a table that stores users in the Elephant SQL database:

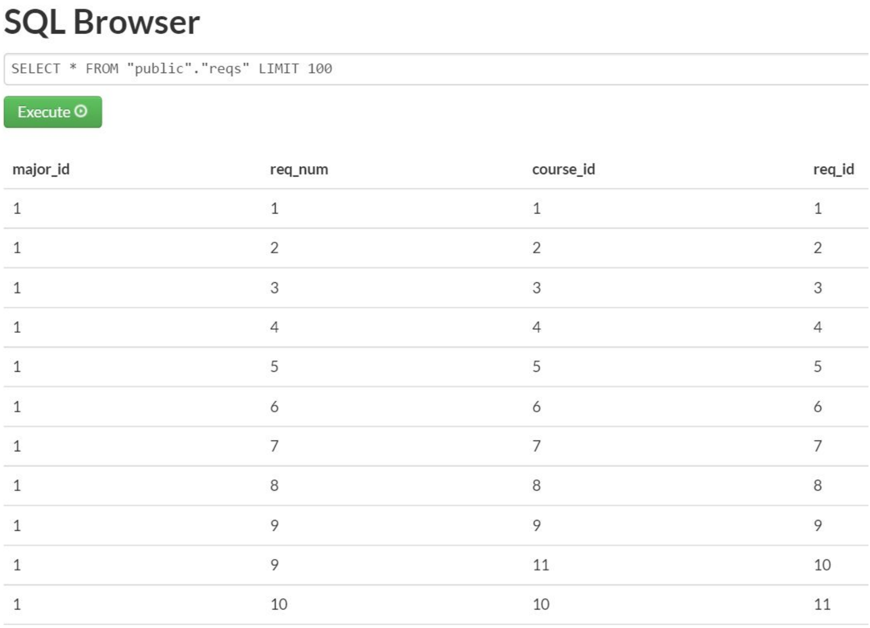


SQL Browser

```
select * from users
```

username	pass	admin	id
test	test	false	1
test2	xxx	false	6
admin	admin	true	2

There is also a lot of cross referencing between the various tables in our database. For example, our user_courses table relies on references to both the user and course_list tables to create a persistent mapping between users and the courses they have added. Reqs works similarly, by referencing the course_list and major tables. This is an example of a table that references other tables:



SQL Browser

```
SELECT * FROM "public"."reqs" LIMIT 100
```

major_id	req_num	course_id	req_id
1	1	1	1
1	2	2	2
1	3	3	3
1	4	4	4
1	5	5	5
1	6	6	6
1	7	7	7
1	8	8	8
1	9	9	9
1	9	11	10
1	10	10	11

For our routes, we have user-routes, major-routes, courses, and admin-routes to direct users between different pages. The app uses the HTTP methods post and get to display requested information and do requested actions. Request and response objects are used for maintaining statefulness throughout the various routes and paths. By keeping track of the user sessions, our routes are able to check for the validity of the user when processing requests. The routes use the user object to check if a visitor is able to access certain views based on whether they are logged in and what privileges they have. This is all accomplished through the express-session library, which is used frequently in our routes. We also use libraries like the body parser to extract information about the user's input.

Conclusion

Overall getInSPIREd was a success. We created a “getInSPIREd” button that students could use to find out what concentration they were closest to completing. Another option students had was that they could enter a class they really liked, and the getInSPIREd page would tell them what concentrations/majors require this class. We believe this could be really useful to students who are unsure of what they want to major in, but know they really enjoyed a few courses. getInSPIREd gives students better perspective of how far they are to completing their major and possible other majors which is not found on SPIRE. We believe that our page is also easier to use than SPIRE. Class searches are much easier, and including ratemyprofessor.com information can help students decide what class they will take going forward.

We originally envisioned creating a website that would be able to handle any majors but quickly realize we would have to narrow our focus, we decided to work with just Computer Science and Math majors for now. Unfortunately we also did not include GENED requirements and other requirements made by the University. In the future it would be great to expand the database to include information about all majors. Another idea we needed to abandon was an algorithm that could effectively schedule potential schedules for users going forward. We found that there were too many variables and wanted to focus on making the website very good at less things than average at many things. Going forward it would be great to see the project learn a scheduling algorithm that could really help students get organized with their studies. We also were not able to create graphics that could represent how close students were to completing their major. There is a pie chart on the profile page to show what this might look like.