COSI 125a: Human-Computer Interaction

Term Project

RateMyCourses

A Collaborative Course Evaluation Environment

Authors:
Stephen Gelman
Ravi Kotecha
Andrew Lewis
Kevin Weaver

Professor:
Richard Alterman

Teaching Assistant:
Alex Gaman

Prototype 1: `http://ratemycourses-dev.brandeis.edu`
Prototype 2: `http://ratemycourses.brandeis.edu`

5 May 2010

Abstract

In this paper we present RateMyCourses, a collaborative course evaluation and social tagging environment. The aim of the site is to fully develop an asynchronous collaborative knowledge community based on collective experiences of students in the classroom. The scope of this utility is within Brandeis University and includes students, faculty, staff, and administrators. In the design of RateMyCourses we have taken a user-centric approach. We design iteratively, seeking frequent user feedback. Here we outline, in detail, the design process and analysis during and after the development of RateMyCourses 2.0. Finally, we identify both strong and weak points in our interface design and enumerate several areas that can be improved upon in the future.

Contents

Chapter 1

Introduction

For our term project this semester we built upon the RateMyCourses (RMC) system developed by Andy Lewis, Ravi Kotecha, and Stephen Gelman in CS118. We added Kevin Weaver to the development team this semester to build upon the prototype we developed last year. RMC is a collaborative course evaluation and social tagging environment. The aim of this site is to fully develop an asynchronous collaborative knowledge community based on collective experiences of students in the classroom. This website builds relationships between courses across academic departments by linking classes with tags through keywords. Courses can also be directly linked to one another, something we call "Cross-linking." The scope of this utility is within Brandeis University and includes students, faculty, staff, and administrators. People who use the website can see all the courses offered at Brandeis and tag them with various tags relating to the courses, such as "Writing Intensive", "Lots of Work", etc. Tags can also relate to course topics, providing the ability to match up similar courses between departments. Students are also offered the opportunity to rate courses they have taken. Also, similar to sites like Amazon.com, RMC will keep track of what users have rated and display a box similar to "Students who took this course also took..." so that students can find similar courses. Our goal with this project is to give students a better idea of the courses offered at Brandeis and to enable them to know more about the content and quality of a class going into it.

1.1 Current Approaches

Brandeis currently provides two ways to search for classes. The first is through the Registrar website, which filters the course schedule by subject, description keyword, requirement, and class meeting time. The second method is through the Student Union's Course Evaluation Guide. Brandeis currently collects surveys from students at the end of each semester. The surveys rate different aspects of each course, such as, the overall quality, the professor's attentiveness, and the difficulty. At the end of each year, a student is hired to analyze the data and compile summaries and reviews of each of the courses. This method works if students would like to read a summary of a given course during a specific semester. However, the either system does not allow for easy browsing and there is no sort of tagging to allow a student to find similar courses from different departments. The two websites to browse for courses are the Registrar search (http://www.brandeis.edu/registrar/schedule/search) and the Course Evaluation Guide (https://sys.brandeis.edu/ceg/index).

The system we built allows members of the Brandeis community to explore the course curriculum in a new, innovative way based on collaborative tagging. This tool is designed to facilitate course selection and provide students the ability to find classes in different departments based on their interests that they would not have been able to find otherwise.

Though functionality is important to the system, a key component of the website is the interface design. As we have learned in our class, interfaces must be user-oriented in order for people to easily use the website. Additionally, the design process must be a cycle which incorporates end-user feedback throughout the process. Throughout this project lifecycle, we iteratively developed two prototypes, consulting with end-users both during and after development of each. We interviewed users and collected responses, which helped shape the design. After the development of the second prototype, we further applied principles we learned in class, critiquing the interface based on the principles of Norman, Schneiderman, and Nielson and conducting a GOMS analysis on our design.

1.2 Implementation Details

As with version 1.0, we programmed RateMyCourses using a Python web framework called TurboGears. TurboGears uses the Model-View-Controller pattern of software engineering. This means that the data storage aspect is separated from the backend code which is, in turn, separated from the templates that the user sees. The advantage of this separation is that it is easy to change one piece of the application without having to modify everything. This was very useful when we were receiving feedback from people because it made it easy to change the design of the site without having to rewrite much code other than HTML.

The login for the site uses Brandeis' Cosign SSO for authentication (viewable through: http://login.brandeis.edu). The advantage of this is that users can log in with their UNET IDs and passwords. This is useful because it prevents the user from having to register their usernames. The downside to this was that it prevented users from choosing anonymous usernames because most users' UNET IDs are some permutation of their real name or searchable in the Brandeis People Directory. To counter this we added the ability for a user to set an anonymous alias. The advantage of this is that it gives users pseudo-anonymity, but at the same time if a user posts a disparaging comment an administrator can look up who they actually are.

Chapter 2

User Analysis

2.1 User Profiles

Our potential users are students, faculty, staff and administrators of Brandeis University. We determined the needs and requirements of users by a combination of informal interviews and observations, and set upon developing the following user profiles and tasks which cover the majority of users of our system.

User type:	Advanced User "Alice"
User description:	Alice has used RMC before and understands the usefulness of user-defined tags.
Use case scenario:	Alice has recently completed a course, "COSI 125a: Human-Computer Interaction". She recalls that it was a fun course, but it required a lot of reading, and she wants future students to know.
User Tasks:	Alice knows that a tag called "FunClass" already exists, so she selects it from the "Add Tag" menu. She then proceeds to type the word "reading" into the "New Tag" text input field. To her surprise, the tag "LotsOfReading" already exists and autocompletes as she types. The two new tags appear on the course page immediately after they are added.
Needs:	Must be able to define a new tag. Must be able to search for existing user-defined tags.

User type:	Novice User "Bob"
User description:	Bob is less likely to add a tag to a course himself, but he understands the usefulness of tags.
Use case scenario:	Bob logs in to RMC and finds the course page for COSI 125a. He agrees that it was a fun course, but found the amount of reading to be quite reasonable, so he disagrees with Alice's choice of the "LotsOfReading" tag.
User Tasks:	Bob upvotes the "FunClass" tag and downvotes the "LotsOfReading" tag by clicking the corresponding arrows next to the tag text.
Needs:	Must be able to vote on the usefulness of a user-defined tag.

User type:	New User "Carol"
User description:	Carol has just started to use RMC and has few preconceived notions about tagging in general.
Use case scenario:	Carol is looking for a fun course to take in the Computer Science department.
User Tasks:	Carol logs into RMC and enters "Computer Science fun" into the search bar at the top of the page. In addition to the tags "Computer-Science" and "FunClass", the search reveals a list of courses that contain both of these tags.
Needs:	Must be able to do a keyword-based search for a course. Must be able to understand the usefulness of tags at a glance.

User type:	New User "Dennis"
User description:	Dennis is not exceptionally computer-savvy, but also dislikes reading directions.
Use case scenario:	Dennis has just logged into RMC for the first time and is unsure of where to start.
User Tasks:	On the home page, Dennis follows the tutorial for first time users. This provides a nice visual walkthrough with accompanying text. Dennis does so, learning how to search for and tag courses in the process.
Needs:	Must be able to know what to do upon logging in for the first time. Must be able to quickly learn how to rate and tag courses.

User type:	Experienced User "Emily"
User description:	Emily has used RMC before and has rated a few courses. She has not necessarily tagged any courses or voted on any tags.
Use case scenario:	Emily intends to log in to RMC today to find some courses for the next semester that might interest her.
User Tasks:	After logging in to RMC, Emily notices a message at the top of the page listing a few courses that she might enjoy based on her previous ratings.
Needs:	Must be able to find recommendations for courses to take in the future.

2.2 Requirements

Based on these tasks, we identified the following requirements:

Requirement id:	1
Requirement name:	Define New Tag
Requirement type:	Functional
Description:	Users must be able to define new tags and apply them to courses.
Rationale:	RMC is built around the idea of folksonomy tagging, which requires giving users the power to create new tags.
Source:	User
Dependencies:	None
Conflicts:	None
Supporting materials:	None

Requirement id:	2
Requirement name:	Search for Tag
Requirement type:	Functional
Description:	Users must be able to search for previously-defined tags with ease.
Rationale:	In order to avoid defining redundant tags, users should be able to search for and quickly identify tags and tag definitions that may fit their needs.
Source:	User
Dependencies:	None
Conflicts:	None
Supporting materials:	None

Requirement id:	3
Requirement name:	Vote on Course Tag
Requirement type:	Functional
Description:	Users must be able to vote on the usefulness of a user-defined tag for a given course.
Rationale:	Because any user can apply any tag to any course, there needs to be a way to democratically determine the tag's appropriateness and usefulness.
Source:	User
Dependencies:	None
Conflicts:	None
Supporting materials:	None

Requirement id:	4
Requirement name:	Search for Course
Requirement type:	Functional
Description:	A user must be able to quickly search for a course based on the course name, description, or tags.
Rationale:	Users will not be willing to spend a long time searching for a particular course in order to rate or tag it.
Source:	User
Dependencies:	None
Conflicts:	None
Supporting materials:	None

Requirement id:	5
Requirement name:	Understanding Tagging
Requirement type:	Usability
Description:	New users must be able to quickly understand the usefulness of tags.
Rationale:	Without a decent understanding of folksonomy tagging, a user cannot properly evaluate a given course, and the site is rendered useless.
Source:	User
Dependencies:	None
Conflicts:	None
Supporting materials:	None

Requirement id:	6
Requirement name:	New User Orientation
Requirement type:	Usability
Description:	A new user must be unobtrusively guided through the site.
Rationale:	A user should not be forced to explore the site in order to learn how it works. New users should be gently guided through rating and tagging their first courses in order to help them develop an understanding of RMC.
Source:	User
Dependencies:	None
Conflicts:	None
Supporting materials:	None

Requirement id:	7
Requirement name:	Course Recommendations
Requirement type:	Functional
Description:	A user must be able to see recommendations for courses to take based on past ratings.
Rationale:	The usefulness of RMC is based on its ability to assist users in finding interesting courses. A recommendation engine would take some of that work out of the users' hands and give them an incentive to continue rating and tagging courses.
Source:	User
Dependencies:	None
Conflicts:	None
Supporting materials:	None

2.3 Task Analysis

We created a hierarchical task analysis for our tasks which helped determine the design and layout.

Hierarchical task analysis for Task-1: Define New Tag

1. Go to a specific course page
2. Click “Add Tag”
3. Choose tag from list OR define new tag with description

Hierarchical task analysis for Task-2: Search for Tag

1. Enter keyword into search box
2. Hit return key
3. View results

Hierarchical task analysis for Task-3: Vote on Course Tag

1. Go to a specific course page
2. Click thumbs up or thumbs down button next to tag

Hierarchical task analysis for Task-4: Search for Course

1. Enter keyword into search box
2. Hit return key
3. View results

Chapter 3

Conceptual Model and Interaction Style

Like many social bookmarking sites, the underlying conceptual model of RateMyCourses is a large, directed graph representing relationships between different nodes. The three main node types are courses, ratings, and tags.

Course nodes are created by the system via feeds from the Registrar. These nodes are static and are not modifiable by users, so the interface must clearly convey this lack of affordance. Courses must be easily searched by users; for this, we provide a simple keyword-based search, as well as a more advanced tag-based search.

Rating nodes are directed at a single course. These nodes are created by users who give a course a rating on a five-star scale. While a rating applies to a particular course, a single course may have many ratings by many different users. These are aggregated into a single average rating for the course, which is displayed on a five-star scale at the top of the course page. However, at the moment we do not clearly convey how many ratings have been applied to a single course, and it is not immediately clear that the star rating shown is an average rating.

Tags can be directed at many courses. These nodes are separated into two categories: system tags and user-defined tags. System tags are used to indicate university requirements, major requirements, and other course attributes defined by the Registrar; they cannot be modified by users. User-defined tags are folksonomy tags that are used to describe more qualitative attributes of a course. These tags are not separated stylistically, but are instead separated spatially on individual course pages. System tags are listed

under the course title and are surrounded by horizontal rules, whereas user-defined tags are listed under the course description and are followed by an "Add Tag" link. The rationale behind the spatial but not stylistic separation is to make it clear that all tags can be used to find courses, but some tags cannot be modified.

Additionally, we have implemented vertices known as "crosslinks" that connect different course nodes. This allows users to link two similar courses without the need to create a new tag and apply it to both courses.

Chapter 4

Design Process

At the beginning of the planning process, we met several times to discuss potential stakeholders. We determined that potential users would include primarily students, but also faculty advisers and department administrators. Because this was a continuation of a previous project, we had the advantage of using existing user data from our first prototype. After making a few mainly aesthetic changes to the original site, we carried out several unstructured interviews to get a better idea of what users were looking for. Additionally, we imagined some scenarios based around several typical user profiles and listed a number of needs and requirements that we would need to act on during development.

Using this data, we developed a second prototype that attempted to address several concerns users had about the first prototype, mainly the way courses were searched for and the way different tags were presented and distinguished from one another. We also decided, based on time constraints, to eliminate some features and ideas that we had, and chose rather to focus on a small subset of features but develop them as best we could. We performed a GOMS analysis on the new prototype in an attempt to identify ways to reduce memory load when searching for and tagging courses.

Once our second prototype was completed, we developed a short survey and distributed it to a wide audience with the intention of soliciting feedback on users' first impressions of the new interface. We also interviewed a few more people using the same questions from the first interview so that we could identify what aspects were improved and where things had become more confusing.

Chapter 5

Prototype 2 Task Narrative

5.1 Defining a new tag and adding a tag

To define a new tag, the user first logs in. Then, they find a course they would like to tag either via searching or browsing the course list. From the course page, they click the "Add Tag" button, which pops up the window in Figure 5.1. From that window, the user has two choices. First, they can type a tag into the text box. As the user types, the box will attempt to autocomplete from the list of tags already in the system. If the user enters a tag that does not exist the tag is added to the system when they click "Add Tag". Alternatively, the user can click on a tag in the list of tags to add it to the course.

5.2 Searching for a course

There are two ways that the user can find a course. One way is by using the search box. If the user types part of a course name in the search box in the upper-right corner of the page, the system will display a page as in Figure 5.2. The user can then select a course from that list. Alternatively, the user can click the "Courses" link at the top of the screen to browse for a course. The course browsing page is a tree view that separates the courses by subject (see Figure 5.3). Clicking on a "folder" for a subject expands it and shows the courses within it. Then, clicking on a course name brings the user to the course page.

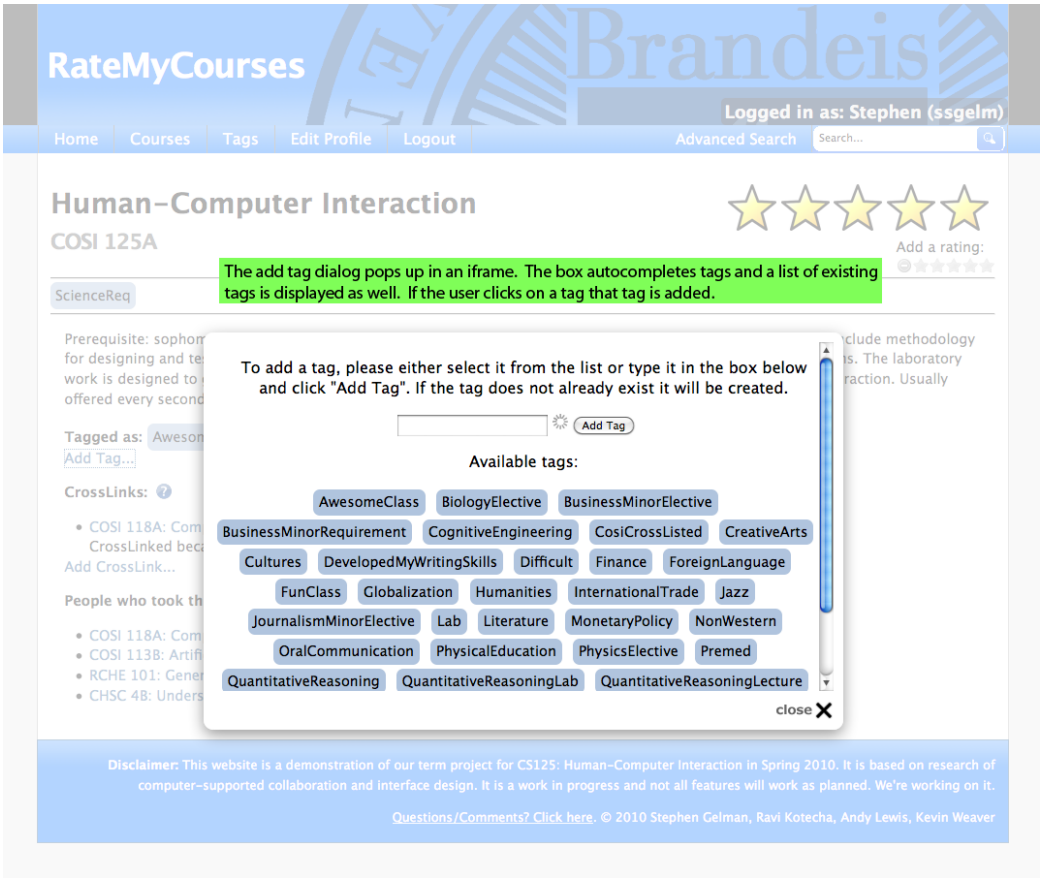


Figure 5.1: Adding a tag

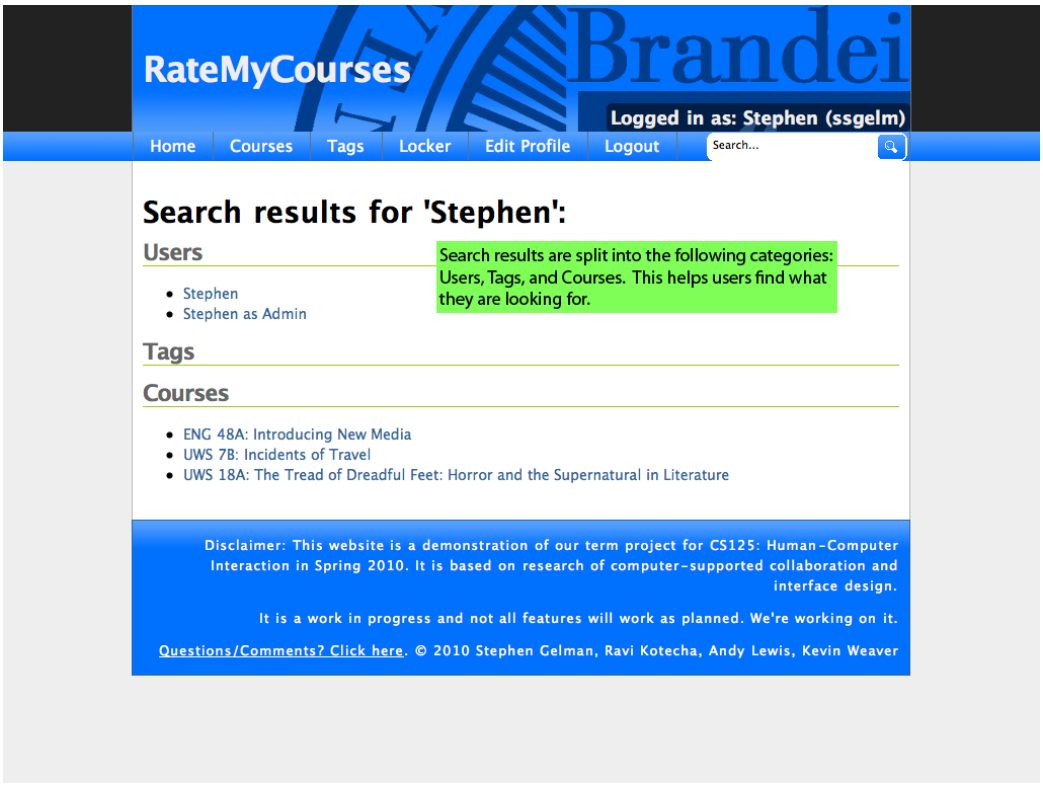


Figure 5.2: Searching by keyword

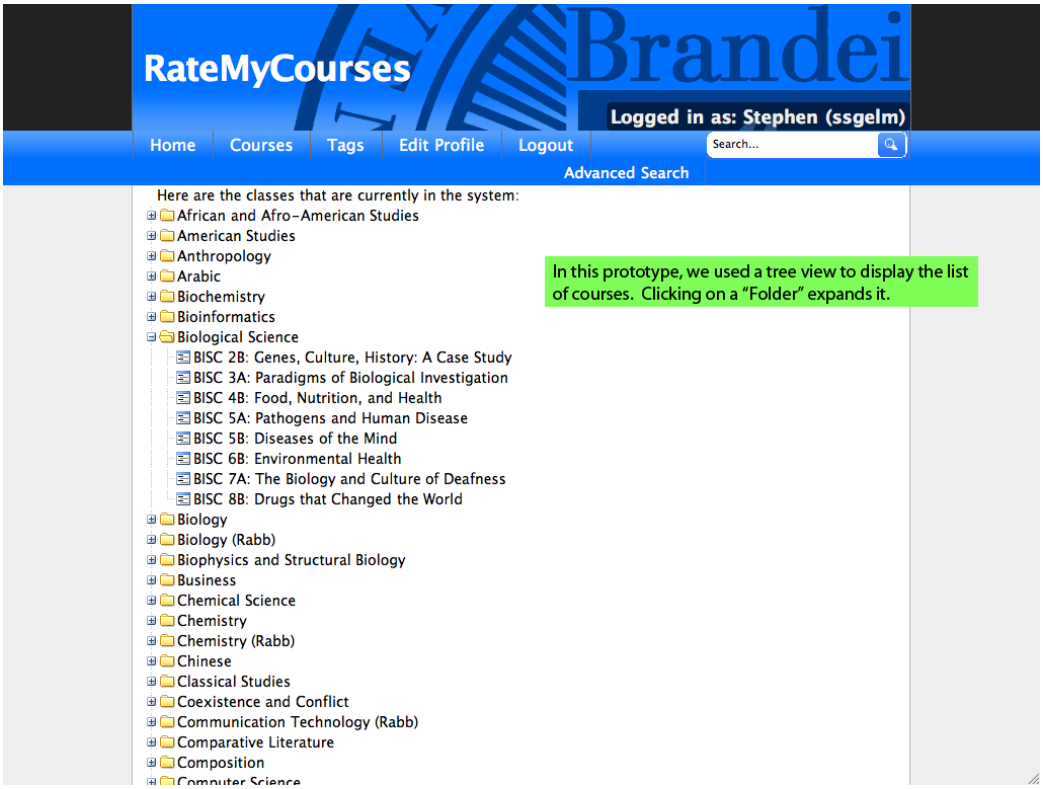


Figure 5.3: Searching by hierarchical treeview

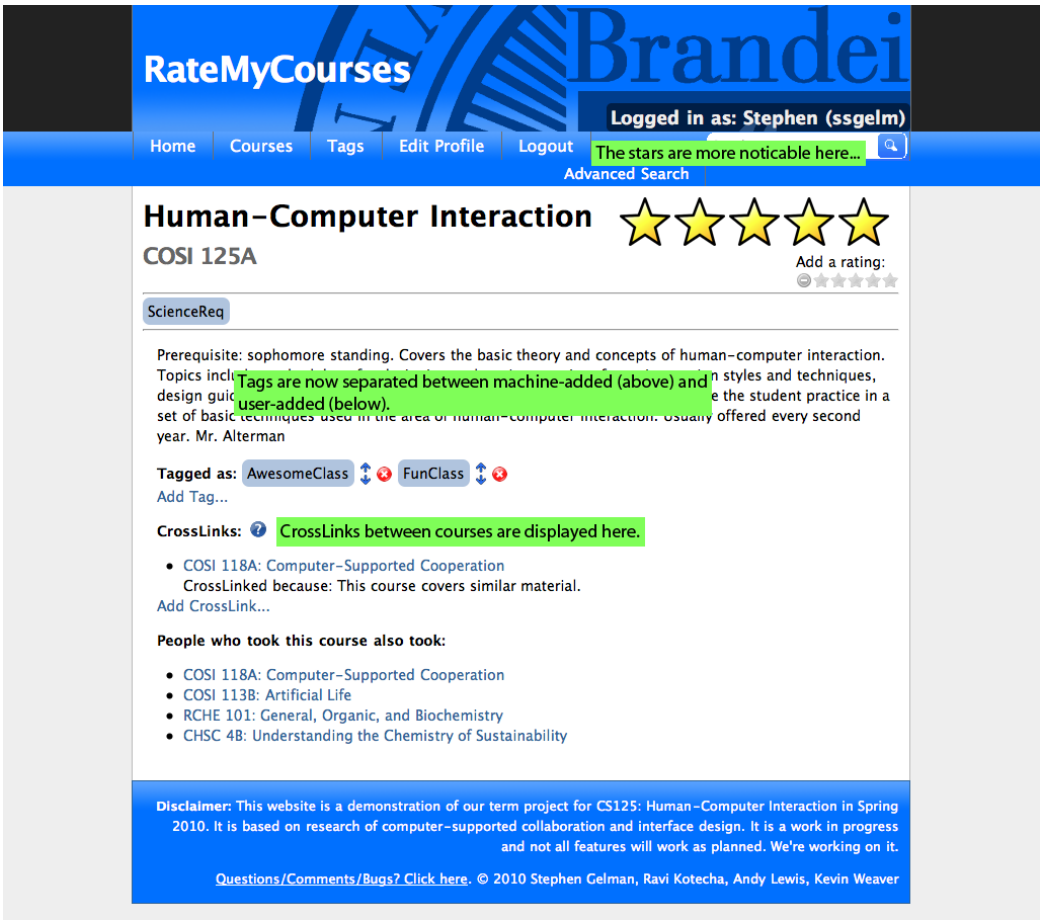


Figure 5.4: Voting on a course

5.3 Voting on a course

Voting is done from the course page. To vote, a user clicks on the up or down arrows next to a tag (see Figure 5.4). Voting can only be done on tags that are user-added.

5.4 Adding a Crosslink

Crosslinks are added from the course page as well. To add a Crosslink, the user clicks on "Add Crosslink...". A dialog pops up as in Figure 5.5. From

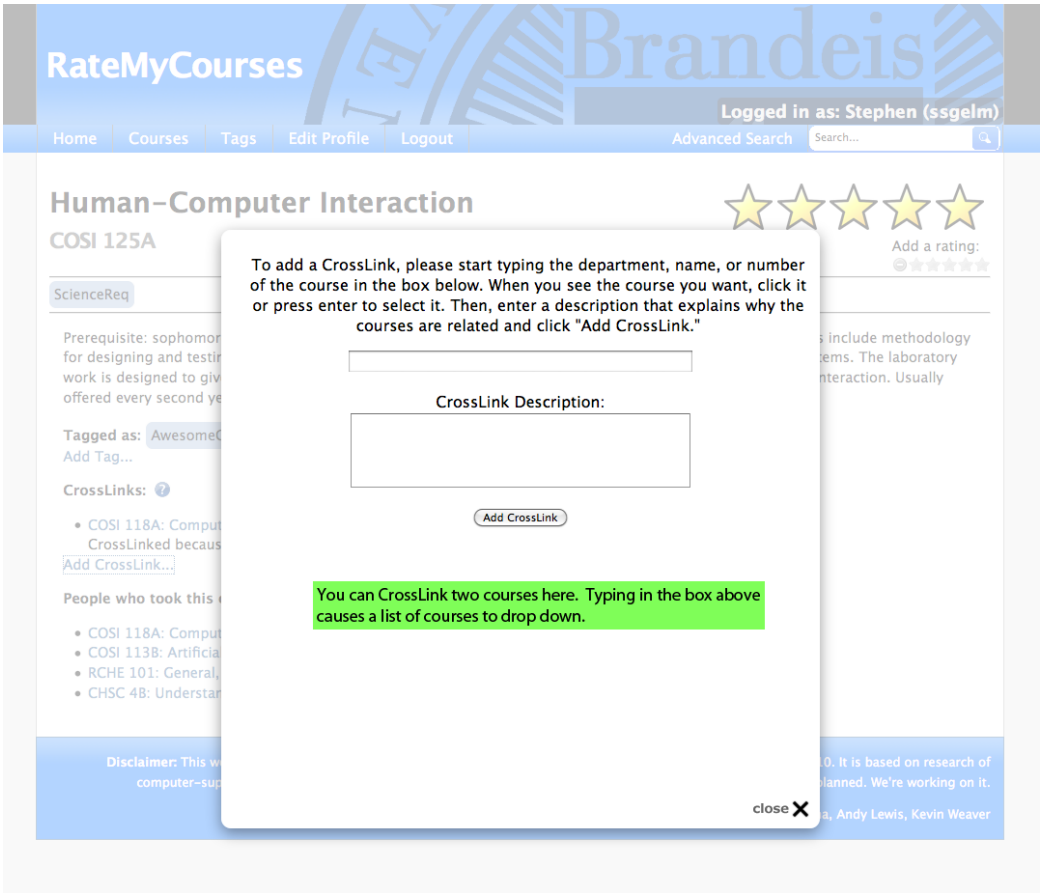


Figure 5.5: Adding a Crosslink

the add Crosslink dialog, the user can begin typing the name of a course and it will autocomplete it. In addition, the user must enter a description of the Crosslink so that it has meaning.

Chapter 6

Analysis, Conclusion and Future Improvements

6.1 Analysis of Survey Data and Prototype Walkthroughs

We designed a survey to distribute after the completion of prototype 2. We received 39 responses over the course of three days. Full survey response data can be found in the appendix. We began by asking general questions about their university status, class standing, and information on how they currently browse courses. From the data we gathered from Brandeis students, we can conclude that generally consult the Registrar's Course Catalog or the Student Union's Course Evaluation Guide a few times per year or semester, presumably around the course selection time. We then directed our questions specifically toward RateMyCourses. A majority of the participants reported that the overall purpose of the site was easily conveyed and for specific tasks such as finding, rating and tagging, the tasks were simple to complete.

The feedback we received from open-ended questions in the survey, as well as the prototype walkthroughs (also in the appendix), provide interesting suggestions, ideas and problems with our current design. Overall, user testing, interviews, GOMS and design heuristics evaluations, reveal that we still have a way to go in terms of error detection and prevention, help, documentation, features and usability. Users brought to our attention ideas and problems that we did not think about, thus highlighting the importance of having the end-user in the design process.

6.2 Conclusion

Overall, we made great progress since we began developing this project in Spring 2009 for CS118. Particularly between Prototypes 1 and 2 this semester, we focused on a few key features, implementing them the best we could, making sure to get feedback from end-users. The interface has become much cleaner and simplified. Aesthetically, we implemented a new CSS layout and completely re-worked the way users add tags (through an iFrame that pops up over the page) and browse the list of courses (tree view). We also added crosslinking and voting on tags, among other features. With Prototype 2 in production for only a few weeks, RateMyCourses holds the following statistics:

- 1891 courses in the database
- 93 users (students and faculty)
- 51 tags
- 98 ratings
- 8 Crosslinks

A majority of the users focused on the rating aspect of courses and a significant number added or created tags. Not many created crosslinks, either because they did not try or we did not clarify the meaning of this well enough.

We were constrained by time and resources throughout the development cycle. Though we spoke with plenty of students, we would have liked to have had time to speak with more professors, staff and administrators. As we learned this semester, design is an iterative process and it is not possible to create the best design in the first attempt. We still have many ideas and visions for RateMyCouses and have seen much improvement through each iteration.

6.3 Future Improvements

6.3.1 Recommendation engine

A major improvement to the RMC web application would be a recommendation engine similar to that of Netflix or Amazon.com. The recommendation

engine would be capable of giving each user a list of courses that they may find interesting. Ideally, it would be based on courses that the user has previously rated or cross-linked. The star rating system, tagged courses, and a way to save courses that have been rated would probably be important for this.

6.3.2 A "Locker" for each user

A "Locker" would allow each user to have a half-private/half-public profile page that included a list of courses they have taken, a list a courses they have bookmarked and wanted to save for later, and also a comments section. The comments section would allow users to get feedback from other users and potentially receive recommendations for each person. Ideally, the recommendation engine would make this piece unnecessary, but computers make mistakes.

6.3.3 Help Section

It would be useful to have a documented help section that explains what things mean on the site, as well as a tutorial for new users. The tutorial would have a simplified "How-to" that would explain how to add tags, add ratings, crosslink, etc. We believe the interface is extremely intuitive, but this would still be helpful.

Appendix A

Prototype 1

A.1 Initial User Interviews

A.1.1 Script Outline

1. What is your university affiliation?
2. If undergrad student, what is your class year?
3. What is your major(s) and minor(s)?
4. What are some important factors when deciding what classes to enroll in?
   (a) Please list factors in order of importance.
   (b) How do you go about specifically choosing what courses to take, based on these factors?
5. How important is it to take classes out of interest rather than to fulfill a specific requirement?
6. How do you find courses related to your interests, but not in your major/minor?
7. How easy is it to find courses or information about courses from the current course guide on the Registrar's website?

A.1.2 Interview 1

1. What is your university affiliation? **Student**
2. If undergrad student, what is your class year? **Senior**
3. What is your major(s) and minor(s)? **IIM: Communication and Media Studies, Minor in Business**
4. What are some important factors when deciding what classes to enroll in? **relevance to major, relevance to career path, schedule, professor, rating from course evaluation guide, workload rating**
   (a) Please list factors in order of importance. **relevance to career path, relevance to major, professor/rating from the course evaluation guide, schedule, workload rating**
   (b) How do you go about specifically choosing what courses to take, based on these factors? **I pick my required courses first then I determine next courses based on what will work with the timing of the essential courses. Then I decide what courses look the most interesting that work with that schedule.**
   (c) How do you decide what classes look the most interesting? **I look in departments that I find the most interesting or that I know I have a history of enjoying classes from those departments. Then I look for the most interesting title and read the summary, then review the course and professor rating.**
5. How important is it to take classes out of interest rather than to fulfill a specific requirement? **Its very important because obviously you need to take the required courses, but being able to take classes that spark your interests is one of the greatest benefits of being a college student. It offers a unique experience that you wont be able to get in the rest of your life.**
6. How easy is it to find courses or information about courses from the current course guide on the Registrar's website? **Information about**

courses could be more detailed in terms of the description. It might be too short of a description. I feel it could use another paragraph or two because it is often vague. The navigation of the overall Registrars site is good though.

A.1.3 Interview 2

1. What is your university affiliation? **Student, undergrad.**
2. If undergrad student, what is your class year? **Senior.**
3. What is your major(s) and minor(s)? **Politics and Economics, with a Legal Studies minor.**
4. What are some important factors when deciding what classes to enroll in? **Course time, professor, major/minor requirements, university requirements, interest.**
   (a) Please list factors in order of importance. **Time, major/minor requirements, university requirements, interest, professor.**
   (b) How do you go about specifically choosing what courses to take, based on these factors? **Go through the schedule looking for courses to fill major/minor requirements, then search for ones with good times, then check description for interest, then check the professor.**
   (c) How often do you use the website each semester? **Check a few times during registration, then a few times during the add/drop period.**
5. How important is it to take classes out of interest rather than to fulfill a specific requirement? **It's important, if you take only required classes, that means you are not expanding your horizons, and it can get pretty boring. I've taken a few courses that fulfill no requirements because I needed to try something different.**
6. How do you find courses related to your interests, but not in your major/minor? **Word of mouth (recommendations from friends), and I know some of my general interests (e.g. chem, cosi).**

   (a) Is there useful info in registrar's site to find these courses? **No, because you cannot search based on the course description.**

7. How easy is it to find courses or information about courses from the current course guide on the Registrar's website? **Fairly easy, most difficult task is finding one more course based on interest (after filling schedule with required courses).**

8. Are there any changes to the registrar's site or CEG that you would like to see? **I would like to see scores for courses grouped by major (e.g. see how politics majors rated a chemistry course).**

A.1.4 Interview 3

1. What is your university affiliation? **Student**

2. If undergrad student, what is your class year? **Junior**

3. What is your major(s) and minor(s)? **Studio Art and Classical Studies**

4. What are some important factors when deciding what classes to enroll in? **major requirements, interest level based on descriptions for classes in other departments, scheduling**

   (a) Please list factors in order of importance. **major requirements, interest level based on descriptions for classes in other departments, scheduling**

   (b) How do you go about specifically choosing what courses to take, based on these factors? **majors - the two majors, then history, english pscychology what needs to be taken and then whatever works best**

5. How important is it to take classes out of interest rather than to fulfill a specific requirement? **the requirements have to be done but emphasis is placed on classes of interest as long as requirements are also being met**

6. How do you find courses related to your interests, but not in your major/minor? **just look through all of the majors, mostly humanities ones and read the titles/descriptions and also cross-listed courses from the taken majors**

7. How easy is it to find courses or information about courses from the current course guide on the Registrar's website? **it's easy to search based on requirements**

8. Would it be useful to search other departments more easily to find related courses? **Yes**

9. Friend recommendations? **Not that important, but could be a factor in deciding between two courses**

10. Are there any changes you would like to see to the registrars site? **See classes from other semesters, when the last time it was offered would be useful**

11. Are you familiar with the CEG on the Student Union website? **Wait they do that? Hmm I might have used it freshman year but I didn't think that it was that useful.**

A.1.5 Interview 4

1. What is your university affiliation? **Student**

2. If undergrad student, what is your class year? **Senior**

3. What is your major(s) and minor(s)? **Chemistry, History (minor)**

4. What are some important factors when deciding what classes to enroll in? **requirements, interests**

   (a) Please list factors in order of importance. **requirements, interests**

   (b) How do you go about specifically choosing what courses to take, based on these factors? **Look at the course list, start with Chemistry and go from there, department by department**

5. How important is it to take classes out of interest rather than to fulfill a specific requirement? **brings excitement into life, requirements are prioritized**
6. How do you find courses related to your interests, but not in your major/minor? **Start looking through departments based on opening in the schedule, perhaps use the search by time**
7. How easy is it to find courses or information about courses from the current course guide on the Registrar's website? **it's fine**
8. Would it be useful to search other departments more easily to find related courses? **Yes. "if you like this, you might also like"**
9. Friend recommendations? **Depends on the friend**
10. Are there any changes you would like to see to the registrars site? **No.**
11. Are you familiar with the CEG on the Student Union website? **Yes, doesn't use it because he is lazy**

A.2 Cognitive Walkthrough

A.2.1 Task 1: Navigating to an individual course page

1. Define Inputs to the walkthrough
   (a) Identification of the users
      i. Students and faculty at Brandeis
   (b) Action sequences for completing the task
      i. Click "Login"
      ii. Then click "courses"
      iii. Either find a class by using the page numbers or select a department from the drop down menu.
      iv. Click the link for the desired course.

   –OR–

      i. Click "Login"

ii. Then type in a search query in the search box
iii. Press enter to search.
iv. Click on the desired search result.

(c) Description of implementation of the interface

i. Users can find a class by either browsing the course listing, or by searching by keywords.

2. Convene the analysts

3. Walk through the action sequences for each task

(a) Tell a credible story, considering

i. Will the user try to achieve right effect?

A. The naming convention should make it clear that a course can be found by clicking the courses button or by searching. It is assumed that a user will know that they have to find a course in order to view a description, add a tag, or do any other function related to finding the course.

ii. Will the user notice that the correct action is available?

A. If the user wants to find a course, they obvious link is the one that says "Courses." However, when they search for one, they might not realize that they can pick by department rather than by going through each page. An issue is that the pages are random. The user may not be able to easily find the right page to find the course they want, which could be frustrating.

iii. Will the user associate the correct action with the effect that user is trying to achieve?

A. If the search string does not match any course names, the search will yield minimal results, if any. Since "Courses" lists a manifest of all the available courses, it is likely the user will understand what will happen if they click that button.

iv. If the correct action is performed, will the user see that progress is being made toward solution to the task?

A. After any link is clicked, another page opens. If they open the course page, they will see the information for that particular course.

4. Record Critical Information

   (a) User knowledge requirements

      i. The user needs to know what courses exist and perhaps their departments if they try to browse by name.

      ii. The user has to understand why they want to find a course.

   (b) Assumptions about the user population

      i. We are assuming people know what departments the classes they have taken are in and that the users go to Brandeis. The website requires a Brandeis user name and password, so we assume they are members of the Brandeis community with a valid login.

      ii. We assume people can get to the website.

   (c) Notes about side issues and design changes

      i. Users may not know what they can search for or how to actually find a course without trial and error.

      ii. A description or some kind of text needs to be added to the front page to inform users what their possible actions are and a help section to tell them how to do it and why.

A.2.2 Task 2: Add a tag

1. Define Inputs to the walkthrough

   (a) Identification of the users

      i. Students and faculty at Brandeis

   (b) Action sequences for completing the task

      i. Follow steps in Task 1 to find a course and then click on it to open the page for that specific course

      ii. Read the tags listed underneath the description of the course to decide if a tag needs to be added

iii. To add a tag, click on the drop down menu to select an existing tag, or click Add new tag at the bottom of the drop-down. o iv. Type in the tag in camel case.

iv. Click OK

(c) Description of implementation of the interface

i. Users will go to the website, and then after logging in, try to find a course and they may decide to tag it. Tagging allows for a way to describe attributes of a particular course and a tag can be shared by multiple courses, which in a way, relates them.

2. Convene the analysts

3. Walk through the action sequences for each task

(a) Tell a credible story, considering

i. Will the user try to achieve right effect?

A. The user may not actually know what the purpose of tagging is and if they select the wrong tag, theres no easy way for the user to delete the tag they just added.

ii. Will the user notice that the correct action is available?

A. The drop down menu right below the list of existing tags says Add a tag.

B. They may not realize that there is a drop down and may think they have to click a link and will see another page to actually select the tag they want.

C. They may also think that Add a tag means to type in their own tag.

iii. Will the user associate the correct action with the effect that user is trying to achieve?

A. There is a possibility that the user will not know how to add a description to a course. There isnt a definition of what tags are and what they can be. This could cause confusion to a beginner.

iv. If the correct action is performed, will the user see that progress is being made toward solution to the task?

A. After the tag is added, it will appear on the course page.

4. Record Critical Information

   (a) User knowledge requirements

      i. The user needs to know how to find courses

      ii. The user needs to know what the purpose of a tag is and why they want to add one.

   (b) Assumptions about the user population

      i. We are assuming people know what departments the classes they have taken are in and that the users go to Brandeis. The website requires a Brandeis user name and password.

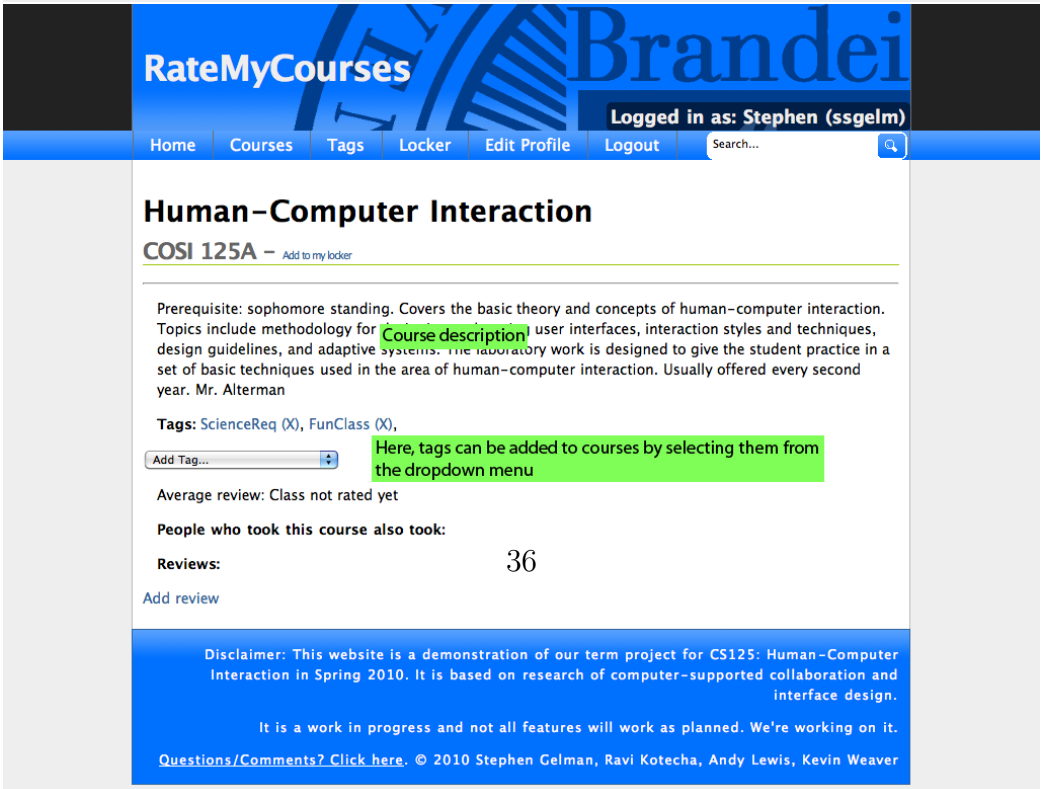
      ii. We assume people can get to the website.

   (c) Notes about side issues and design changes

      i. Make it obvious that the user needs to login before they can tag courses.

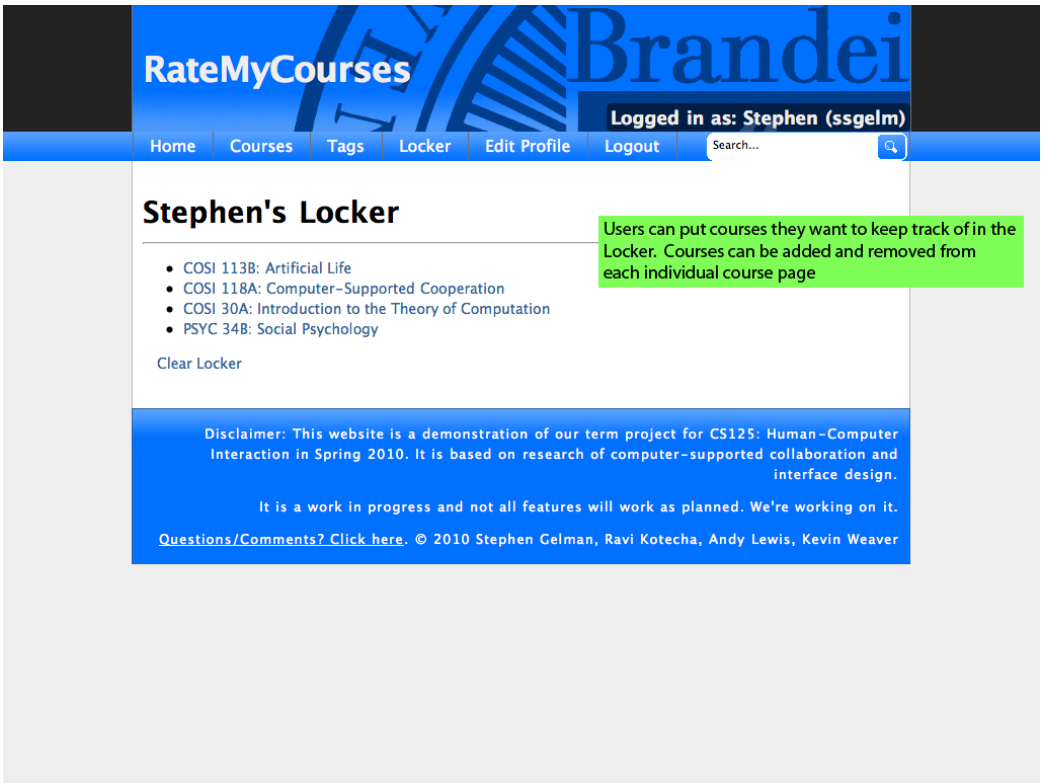
      ii. Change the design of how to add a tag to a course. The conceptual model is not clear. The tags may also need to be categorized so system tags do not appear.

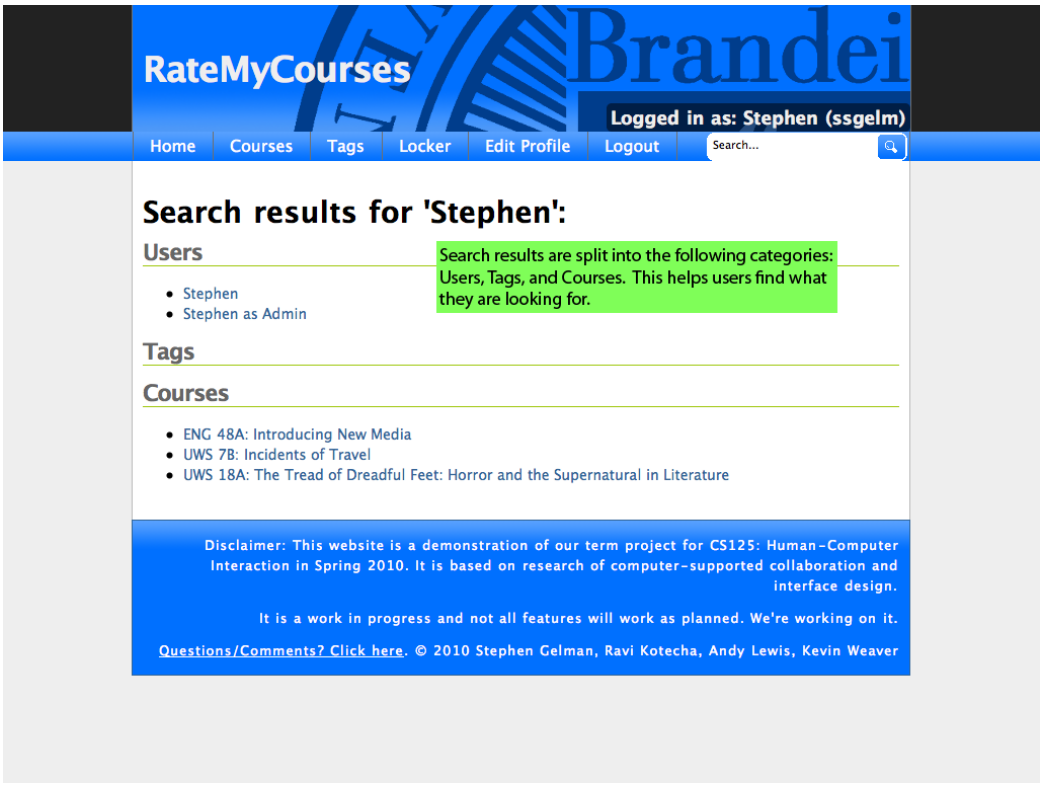
A.3 Screenshots

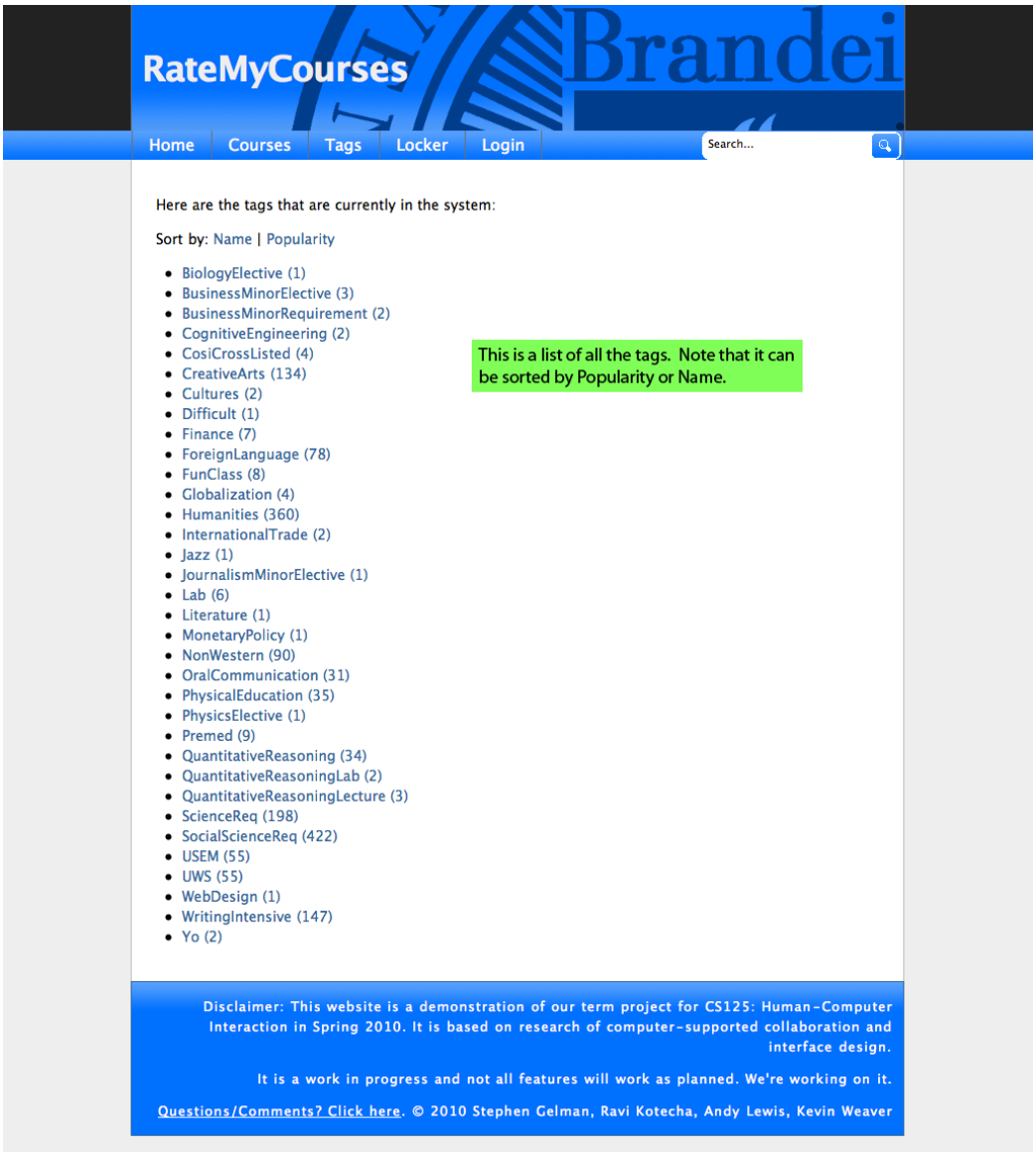



RateMyCourses
Brandei
Home
Courses
Tags
Locker
Login
Search...
Here are the classes that are currently in the system:
Select a Department
One can select a specific department
« ← 1 2 3 4 5 6 7 8 9 10 → »
The list is paginated so it's not one long page
AAAS 80A: Economy and Society in Africa
AAAS 114B: Race, Ethnicity, and Electoral Politics in the United States
AAAS 123A: Third World Ideologies
AAAS 156A: The Civil Rights Movement
AAAS 5A: Introduction to African and Afro-American Studies
AAAS 60A: Economics of Third World Hunger
AAAS 70A: Introduction to Afro-American History
AAAS 82A: Urban Politics
AAAS 85A: Survey of Southern African History
AAAS 115A: Introduction to African History
AAAS 126B: Political Economy of the Third World
Here is the list of courses
AAAS 132B: Introduction to African Literature
AAAS 133B: The Literature of the Caribbean
AAAS 163B: Africa in World Politics
AAAS 171A: Reggae Representation, Race and Nation
AMST 10A: Foundations of American Civilization
AMST 20A: Environmental Issues
AMST 100A: Classic Texts in American Culture to 1900
AMST 100B: Twentieth-Century American Culture
AMST 101A: American Environmental History
AMST 102A: Environment, Social Justice, and the Role of Women
AMST 130B: Television and American Culture
AMST 142B: Love, Law, and Labor: Asian American Women and Literature
AMST 163B: The Sixties: Continuity and Change in American Culture
AMST 105A: The Eastern Forest: Paleoecology to Policy
« ← 1 2 3 4 5 6 7 8 9 10 → »
Disclaimer: This website is a demonstration of our term project for CS125: Human-Computer Interaction in Spring 2010. It is based on research of computer-supported collaboration and interface design.
It is a work in progress and not all features will work as planned. We're working on it.
Questions/Comments? Click here. © 2010 Stephen Gelman, Ravi Kotecha, Andy Lewis, Kevin Weaver


RateMyCourses
Brandei
Logged in as: Stephen (ssgelm)
Home
Courses
Tags
Locker
Edit Profile
Logout
Search...
ssgelm's Profile
Alias:
Stephen
Here you can set an alias to prevent your real name from showing
About Me:
This is my profile.
Here a user can set a profile that appears on their locker page
Submit
Disclaimer: This website is a demonstration of our term project for CS125: Human-Computer Interaction in Spring 2010. It is based on research of computer-supported collaboration and interface design.
It is a work in progress and not all features will work as planned. We're working on it.
Questions/Comments? Click here. © 2010 Stephen Gelman, Ravi Kotecha, Andy Lewis, Kevin Weaver


RateMyCourses
Brandei
Logged in as: Stephen (ssgelm)
Home
Courses
Tags
Locker
Edit Profile
Logout
Search...
Stephen's Locker
Users can put courses they want to keep track of in the Locker. Courses can be added and removed from each individual course page
COSI 113B: Artificial Life
COSI 118A: Computer-Supported Cooperation
COSI 30A: Introduction to the Theory of Computation
PSYC 34B: Social Psychology
Clear Locker
Disclaimer: This website is a demonstration of our term project for CS125: Human-Computer Interaction in Spring 2010. It is based on research of computer-supported collaboration and interface design.
It is a work in progress and not all features will work as planned. We're working on it.
Questions/Comments? Click here. © 2010 Stephen Gelman, Ravi Kotecha, Andy Lewis, Kevin Weaver


RateMyCourses
Brandei
Logged in as: Stephen (ssgelm)
Home
Courses
Tags
Locker
Edit Profile
Logout
Search...
Search results for 'Stephen':
Users
Search results are split into the following categories: Users, Tags, and Courses. This helps users find what they are looking for.
Stephen
Stephen as Admin
Tags
Courses
ENG 48A: Introducing New Media
UWS 7B: Incidents of Travel
UWS 18A: The Tread of Dreadful Feet: Horror and the Supernatural in Literature
Disclaimer: This website is a demonstration of our term project for CS125: Human-Computer Interaction in Spring 2010. It is based on research of computer-supported collaboration and interface design.
It is a work in progress and not all features will work as planned. We're working on it.
Questions/Comments? Click here. © 2010 Stephen Gelman, Ravi Kotecha, Andy Lewis, Kevin Weaver



A.4 Walkthrough Comments

A.4.1 Walkthrough 1

- Likes design, no big paragraphs
- Good searching

  - Would use search box
  - Usage: would type course number if looking for specific course otherwise, would browse using "Courses" tab

- Consistent with Registar's site (dropdown box)
- page layout: questions order of courses (not in numerical order by course number, should be consistent)
- Course pagination make sense with arrows, but unlikely that someone would browse that far
- Dropdown menu does not work at the moment look at COSI 11a
- Likes our importation of Registar's description and requirements
- Figured out usefulness of tags quickly
- Likes "people who took this also liked..."
- Adding a tag to a course
- Searched for "ECON 2a" and did not find course, searched for "ECON" and found it
- Likes "do you find this review useful/inappropriate"
- Tried to add tag, needed to log in, but Cosign authentication redirected back to home page
- Added "Fun Class"
- Questioned allowing everyone to add tags
  - Moderator would be most useful for inappropriate tag, but that could be a full-time job
  - Next best way would be "is this tag inappropriate?" button with reason - I mentioned voting, he thought it would be useful
- Incentive to rate courses: prizes?

A.4.2 Walkthrough 2

Home Page:

- Tag cloud: stands out and very useful. Really shows what kind of information people are looking for. It's a really interesting concept. Very informative visually.
- Nice and simple layout
- Not too many options
- User friendly

Looking for a class:

- Page numbers make it difficult to figure out what letters in the alphabet fall on each page
- Suggestion: change the numbers to letters
- Four-letter codes for departments are not always clear, suggests display whole name of department next to 4 letter code Individual class page:
- Tags seem appropriate
- Consistent course description
- Add Tag vs. New Tag labels in tag list not clear or intuitive
- Created new tag
- Tried to put a space in the tag name caused an error "Handling bad tags:"
- Should be handled by the Registrar

Overall:

- Just as useful as the course evaluation guide
- Impressed

A.4.3 Walkthrough 3

Home Page:

- Meaning of tag cloud not clear
- Layout easy to read, follows Brandeis color scheme of blue and white
- Simple Course Listing:
  - Alphabetical listing by course load
- Arrows navigating page numbers not clear
- Department listing needs indication of what the four letters stand for

Course Page:

- Easy to read
- Clearly marked and labeled
- Course description consistent with Registrar

Adding a tag:

- Tag listing relatively makes sense. Some definitions are not clear.
- User added a new tag
- Enter Tag dialog box not clear about spacing, capitalization, consistency of tag names
- After login, does not return user to current page. Instead he has to re-navigate to where he was and re-add the tag again.

Appendix B

Prototype 2

B.1 Expert Reviews

B.1.1 Nielsens Ten Usability Heuristics

Visibility of System Status RMC Prototype 2 is designed to ensure the visibility of the system status. At any page, the header displays whether or not a user is logged in, and if so says Logged in as: . Additionally, green or red status menus appear throughout the site based on response to user action. For example, after logging in, the system displays a green box saying: You are now logged in as alewis1! Additional menus to ensure visibility of status appear after tagging, searching, and adding crosslinks.

Match between System and the Real World The language used throughout the site is simple, clear and easy to understand. Between prototypes, we made sure to remove technical terms in headers, menus and bodies of individual pages and replace them with non-technical equivalents. For example, course database was changed to course catalog, which reflects the language of the end-user (students, faculty and staff) rather than our individual technical backgrounds. We worked with a student majoring in Media and Communications to help re-write the text across the site.

User control and freedom Since this is a website, a large part of user control and freedom is controlled via the users web browser, using the

back and forward buttons to navigate between their browsing history. Tabs on the header of each page provide a clear way to navigate between the different pages of the site. However, there lacks an undo or redo feature when adding tags, crosslinks, or ratings. There is no way to undo an action here, and any error or mistake must be corrected from the back-end by an RMC administrator.

Consistency and Standards The layout of this prototype follows the conventions of most websites and is easy to navigate. Overall the site uses consistent language with that of the Registrar and Student Union Course Evaluation sites.

Error Prevention RMC provides good error messages only in certain cases. In entering an invalid search term or rating a class one previously rated, the system provides good feedback to the user that the current action is not possible. However, in the case of adding an incorrect or misspelled tag or crosslink listing, after hitting Submit there is no way to reconfirm or undo an action. For example, if I misspell a tag, the only way to get rid of it is to delete it from the back-end database.

Recognition over Recall Performing tasks in RMC relies on recognition rather than recall. In using the system, the user does not have to remember or recall anything. In the courses tab, RMC displays all of the Brandeis courses, categorized by department. In adding a tag, RMC displays all of the tags in the system and a user can choose from the list. In adding a crosslink, the RMC engine auto-completes based on the users input.

Flexibility and Efficiency of Use We included a few accelerators to enhance the use of the system for advanced users, such as the ability to do an advanced search based on multiple tags. Also, when tagging a course, the auto-complete textbox allows advanced users to directly enter a tag name rather than search through the whole list of tags.

Aesthetic and Minimalist Design This site has a simple and minimal CSS layout and overall site design. Sections of an individual page are separated through color, using blue for the header and footer and white for the body. Tags have a distinct look, with black text and a grey background.

Error Recovery Overall, errors and messages are clearly expressed in plain language. Error messages pop up as a red background, while success messages have a green background. The error messages clearly detect and help users recognize errors, however the error dialogues do not always provide clear information on how to recover from the error.

Help and Documentation Though not yet fully implemented, RMC provides help and documentation to the user. On the home page, we provide three scenarios for users to start using RMC: 1) a walkthrough of the site, 2) login, and 3) just browse the site without logging in. Also, individual pages for tags provide a space for tag definition. An example can be seen here: https://ratemycourses.brandeis.edu/tag/Premed. Additionally, there is a quick help feature for the Crosslink function. On any individual course page, a question mark appears next to Crosslink, which is a tooltip that explains the definition of Crosslink.

B.1.2 Shneidermans Eight Golden Rules of Interface Design

Consistency RMC holds a consistent interface throughout the website. Aesthetically, the site uses the same fonts, layout, color and styles throughout each page. The language used is consistent across pages, as well as consistent with the language on the Registrar and Course Evaluation Guide websites.

User Shortcuts Though not yet implemented, through the Home Page RMC can provide a tutorial or walkthrough for beginning users, giving them a demonstration of what the website is about and how to use it. Expert users can simply begin using the site right away. Explanations, through text or tooltips, describe or simplify tasks. We implemented large fonts throughout the site and clean, simple layouts. Overall RMC is well designed to cater to all levels of users.

Feedback Feedback involving interaction with tasks is generally provided through dialog boxes (both success and error messages). Tagging, searching and crosslinking all offer informative feedback as to state of the system. Since this is a website, feedback when interacting with an individual page is provided through mouse-over actions, such as

highlighting, changing the background color of the hyperlink, or fade effects.

Design Dialogs to Yield Closure Dialogs, and tasks, are designed to yield closure. For example, when adding a tag, the beginning sequence is clicking the Add Tag link, the middle sequence is the iframe which displays the dialog box to add a tag, and the end sequence is clicking submit, which displays a dialog box indicating success or failure of the task. Also, on the Courses page, the tree-view list of courses is designed to yield closure, in this case the selection of an individual course.

Error Prevention and Handling RMC does not fully prevent errors. There is no error checking when adding a tag, and for example I can enter random digits, which becomes a tag. In this case, or in the case of misspelled or incorrectly chosen tag, there is no confirmation to ensure correctness or no way to undo this action. These errors have a cascading effect. One incorrect or misspelled tag created by one user turns into an option for the next user.

Easy Reversal of Actions Reversal of actions is easy if users navigate to a specific page by mistake. This can be reversed by using the browsers back and forward navigation buttons, as well as by clicking the appropriate tab in the header (which appears on each page).

Internal Locus of Control Overall RMC does a good job of supporting internal locus of control. Good feedback ensures that users feel that the interface responds to their actions. Interface interactions are designed to be as simple as possible, and any one task involves a short sequence of actions. Information is clearly displayed throughout the site. When adding tags or crosslinks, RMC is designed to minimize the amount of tedious data entry necessary. When adding tags, the iframe lists all the tags in the system at once, or textbox to enter a tag autocompletes using the results in the system. When adding a crosslink, the search box autocompletes using the courses in the system. Overall, all of these features, as well as others, help users feel like they are in control of the interface.

Reduce Short-term Memory Load The RMC interface is designed to reduce short-term memory load, though it could use more work with this.

When adding a tag, the list of tags does not display their corresponding definitions, which requires users to remember the definition. Generally, the definitions are obvious, however in some cases there may be semantic differences in the tag meaning.

B.1.3 Donald Normans Usability Principles

Visibility Actions possible by RMC are clear and visible on each page. In the header of each page contains links to each section: the home page, list of all courses, list of all tags, login, and basic and advanced searches. On the all courses page, the tree view makes visible a lot of information at once; it lists each academic department at Brandeis, and once expanded, each course within that department. The layout of the individual course page visibly displays all possible actions: add a rating, tag a class, and select a crosslink.

Feedback Overall the site provides good feedback through the use of mouse-over actions and dialog boxes.

Constraints Tasks on RMC are designed with artificial constraints. For example, the tasks of rating, tagging or crosslinking are only possible once a user has authenticated via their Brandeis Unet ID. We reduced the need for knowledge, and have users rely on recall through icons and visuals. Ratings, for example, are visually displayed in terms of stars. Through the use of color, tags are displayed as icons.

Mapping Through the use of icons, hyperlinks and text, there exists mappings between intentions and possible actions. Additionally, through immediate feedback from actions, users can always determine the system state. Overall, the system responds to user intentions.

Affordance RMC contains good affordances that take advantage of web conventions, such as clicking on hyperlinks and pushing buttons to submit a form.

Good Conceptual Model Overall, RateMyCourses has a good conceptual model. Because of the heavy use of tagging in Facebook, most college students are already familiar with the concept of tagging. We took this familiar concept and applied it to the Registrars Bulletin. Though

RMC is still in its initial stages and has a very small amount of usage data, it has the potential to become a vital tool in the exploration of classes across departments.

B.2 GOMS analysis

Key:

K type a key

P point to a position on screen

H homing

M mentally preparing for next step

R responding

L login through Brandeis' Cosign system

Logging in through Cosign (http://login.brandeis.edu)

- P H Kx H M Ky K
  - x is the number of letters in the username
  - y is the number of letters in a password

Finding a course

- Click courses tab with mouse
- Read and evaluate the tree-view of courses
- Click on department name
- Click on course
- P M P P

Tagging a course

- Find the course

- Click add tag
  - New tag
  - Type in name of tag
  - Click “Add Tag”
  - Existing tag
    - Find tag in tag cloud
    - Click tag
- New tag:
  - P M P P M P H Kx H P
  - Where x is the number of letters in the tag
- Existing tag:
  - P M P P P

Crosslink

- Find a course
- Click Crosslink
- Move mouse to course field
- Begin typing course name
- Select course from list with mouse
- Click on description field
- Type in description
- P M P P P M P H Kx H P P M Kx

Rating

- Find a course
- Click on star value to add rating
- P M P P M P

B.3 Walkthrough Comments

B.3.1 Walkthrough 1

Home page

- Background shading for tags is good because it gives better separation
- "New to RateMyCourses" box gives better information on how to use the site

Courses Listing

- Tree view display of classes does not display in Internet Explorer
- User found course through Search box

Individual Course Page

- Likes the fact that the requirements are up top right below the title, that is one of the most important factors people look at when choosing classes
- "Course not rated yet" is good because it gives the system state, in that it is not rated
- Tooltip to explain crosslinking is useful in helping define what exactly crosslinking is, as it is a confusing term
- "People who took this course also took" is empty, which is misleading. Make it clear the way the recommendation engine works. This is confusing.

Rating a course

- "Cool!"
- A lag between selecting stars and the results appearing
- Enjoys the green dialog box which explains that the action was completed successfully

Adding a tag

- iFrame does not work in Internet Explorer, redirects to new page
- Directions clear
- Adding a tag with spaces causes site to crash
- Speculates that arrows cycle through tags. Explained real purpose, but participant said the arrows are not clear. Suggests adding tooltip.
- User added a description for the tag to help other users.

Crosslinking

- Likes the auto-suggestion
- Finds the interface intuitive
- Text of "Crosslink Description" not immediately clear

Overall

- Overall much better than the current rating system
- A few bugs, but once these are fixed Brandeis should use this instead

B.3.2 Walkthrough 2

Home page:

- Why are some of these tags bigger?
- What is the point of the tags?

Course page

- No problems with this
- What is the point of a Crosslink?
- Adding a tag is simple and easy to use

General comments:

- Not easy to understand what is meant by a tag or why they are useful
- Why are they called tags and not categories?
- Probably wouldn't use this site

B.3.3 Walkthrough 3

Home page:

- "Wow look at all the things I can click on"
- A bit confusing at first
- Liked "new user" option
- Instinctively used tag cloud to find courses
- Looked for tags that pertained to her, then her major, then a class she liked

Class page:

- First instinct is to rate
- "Can I write something?"
- Liked the "People who took..." courses at the bottom
- Enjoyed rating classes taken
- Liked that requirements were listed for courses
- Thought hovering over question mark was not obvious for Crosslink description
- Wished it was possible to give half star ratings, i.e. 4.5/5 stars.
- Liked feedback on rating
- Wanted to be able to write reviews to understand particular ratings
- More likely to rate a course because it was good than because it was bad
- Would be helpful to know how many people had rated the course

Courses list:

- Found tree view intuitive

- Located a course very quickly

Add tag:

- Tagging really cool idea
- Liked potential for different tags
- Instinct was to use existing tag instead of creating one

Tags:

- Liked to browse for course by tag: "Let's see what classes were tagged with HotProfessor"

Crosslinking:

- Didn't understand the benefit at first, but once explained liked the idea

Search:

- "That was really fast"
- Not immediately obvious that search box exists
- More likely to want to browse
- Wanted to search by course number but it didn't work

B.4 QUIS Survey Data

We received 39 responses given over the course of three days. We designed this survey with a combination of drop-down, open-ended and Likert scale questions.

What is your university affiliation?

Response	Percent Response	Count
First-year student	10.3	4
Sophomore student	25.6	10
Junior student	17.9	7
Senior student	46.2	18
Graduate student	0.0	0
Faculty	0.0	0
Staff	0.0	0

How frequently do you find yourself browsing the course catalog?

Response	Percent Response	Count
Never	2.6	1
A few times each semester	44.7	17
Many times during one or two months out of the year	42.1	16
A few times each month on a regular basis	10.5	4
Every week	0.0	0
I didn't know this existed	0.0	0
Skipped question	-	1

How frequently do you find yourself browsing the Course Evaluation Guide?

Response	Percent Response	Count
Never	28.9	11
A few times each semester	44.7	17
Many times during one or two months out of the year	21.1	8
A few times each month on a regular basis	2.6	1
Every week	0.0	0
I didn't know this existed	2.6	1
Skipped question	-	1

After browsing RateMyCourses, how well do you feel you understand the purpose of the site?

not at all	somewhat well	fairly well	very well	fully	No Opinion	Rating Average	Response Count
5.6 (2)	2.8 (1)	27.8 (10)	25.0 (9)	38.9 (14)	0.0 (0)	3.89	36

Please rate how easy you found it to complete the following tasks:

	1	2	3	4	5	6	Rating Average	Response Count
Finding a course	0.0 (0)	0.0 (0)	0.0 (0)	24.3 (9)	75.7 (28)	0.0 (0)	4.76	37
Rating a course	2.7 (1)	0.0 (0)	2.7 (1)	16.2 (6)	59.5 (22)	18.9 (7)	4.60	37
Tagging a course	2.7 (1)	0.0 (0)	0.0 (0)	13.5 (5)	54.1 (20)	29.7 (11)	4.65	37
Cross-linking two courses	10.8 (4)	0.0 (0)	0.0 (0)	21.6 (8)	18.9 (7)	48.6 (18)	3.74	37

1. I didn't know this was possible
2. Impossible
3. Difficult (required some help)
4. Easy (figured it out after some time)
5. Simple
6. I didn't try

Was there anything you found to be particularly confusing?

- When you search for classes using the advanced search, it seems like they come up in any random order (not in order). Also, when I searched the term "real estate" some classes came up more than once.
- There should be a login button on the home page in one of the top corners in addition to "I've been here before, log me in!" since most people assume that there will be a login link in one of the top corners.
- No.
- There should be more directions
- The purpose of this site
- Perfectly clear all around
- It wasn't initially clear that you could search by department. The homepage has all the funny boxes for various requirements that are fulfilled by a course, but the folder tabs by major were awesome
- I believe the rating should be centered on the page instead of in the corner. As is, it looks more like just a course catalog instead of a rating system.
- No
- Really easy to navigate. Nice work guys.
- No, great site!

Are there any features that you wish RateMyCourses had that would make it easier to use?

- It would be better if once you rated a course, you could change the rating. When I tried to change the rating, it gave me an error message saying "Course already rated."
- You couldn't untag something that you added accidentally
- I would not user RateMyCourses in its current state. Every time a course is taught, it is taught completely differently. In fact, the most important variable in whether a course will be worth taking is the professor, not the course itself. In effect, all of these ratings will suffer from severe missing variable bias. The Course Evaluation Guide allows me to search both by professor and by course, and allows me to look up ratings of one specific offering of a course, which is much more useful to me.

  Additionally, I find the tagging feature to be completely useless. Tagging depends on someone moderating to make sure that tags are consistently applied and that duplicate tags with similar spellings are not made. The most important tags, such as what requirements the course fulfills, can already be easily searched for on the Registrar's site.
- No.
- Perhaps if the courses were further broken down into sections. (Sometimes different teachers teach multiple sections of the same subject.)
- A comments section might be nice. It would probably be more trouble than its worth, but getting more detailed opinions of a class would be helpful.
- There should be an easily accessible list of the best and worst courses.

  Optimally, there should be a way to rate different components of a course and then rank courses by their average scores for each category.
- List of courses by department
- A back button that will return you to the course tree with it expanded to how it was last. A comment section for each page that is not automatically displayed.

- Rate Professors too???
- No
- Include more about professors
- It should show the number of people who voted on a particular course.
- I wish there was a way to comment on the course or to read comments other people had left about the course. Also the number of people who have rated the course.
- If our past course history was available to the site and we could see suggested courses, then cross-linking could be useful...
- There should be ratings for professors and not just courses. You could cross-link the professors to the courses they teach.