Matthew Harris · Kevin Kane · James Weir · Michael Wood

Notezilla : Milestone 7.0

TABLE OF CONTENTS

Product Concept

Our group will be creating Notezilla, a free web-based application for collecting, viewing, and sharing notes among Chico State students. This application will be used initially by Computer Science students at Chico State. They will save and view their notes, their classmate's notes, and notes from previous semesters easily online.

We feel that this service will be of great value to students as more and more of them are taking notes on their computer or scanning them to save or share with classmates. Notezilla will give students an easier way to take, view, and share their notes. Our group has reviewed the academic honesty policy here at CSU Chico and it does not mention anything about note sharing among peers to be prohibited.

Requirements:

- Operating System: Ubuntu Linux
- Apache 2.2.14 web server
- PHP 5.3.2 interface
- MySQL 5.1.41 database
- Clients will use Mozilla Firefox (initially)
- Cookies will be required to be enabled by the client
- Reliability is dependent on our Linode server
- Very scalable in media taken and size of userbase (all CSU Chico, all CSU's, etc)
- Value of service will scale with number of users
- No personal info collected (other than CSU email address)

Feature Summary:

Account Services

As a web-based application, uses will be required to login to access the site's content. We will not be providing notes to the general public. Upon the user's first visit to a site they will be able to **register an account**. This will ask for their CSU Chico email address, as well as some basic information (name and password). We will send an automated message to their address to verify their account.

Subsequent visits to the site will allow the user to **login to their account**. A cookie will be stored on the user's machine keeping track of their credentials. When the user is done using the site they may **log out**. Closing the browser will also cause the cookie to expire (thus logging them out).

Note Submission

Once logged in, users will be able to **create a note**. They can do this from an individual class page, or directly from a link on the main navigation. Users may **edit a note** that they have created, to correct errors or improve the quality of their submission. Finally, users may also **delete a note** that they have created.

Note Retrieval

By itself, adding notes to the site is meaningless. In order to be a true data-sharing site, users must also be able to **view a note.** Notes can be retrieved in two different ways. First, there will be a catalog page that will list all classes our site supports. Users can **view a class**, which will load all class notes for the current term. They can then choose to **filter notes** to limit what is displayed on the screen. The second way that notes may be retrieved is from the user's home page (which is addressed below).

When a user has retrieved a note, several actions may be performed (in addition to just reading it). If the user finds inappropriate material, such as profanity or non-academic content, they will have the option to **flag a note.** They can also provide a reason for the flag if they want. On the contrary, if a user finds a note that they like they can choose to **add it to 'favorites'**. If they find that another user is consistently providing valuable notes they can also choose to **follow an author**. This will essentially mark all past and future notes from the author as favorites for the user.

Account Home Page

Every user on the site will have a home page. This will be the central hub for the user, and act as their starting page (after login). Their home page will be both a note index as well as a control panel for the user's account. The user will be able to **change their password** from this page, as well as any future account attributes. Faculty will also have the option on their home page to **block classes** that they have been assigned (by our developers).

The user's home page will be a link index, allowing them to **view their created notes.** Further down the page they can **view their favorites** that they've saved. If they decide they no longer want those notes saved they can chose to **remove a favorite**. Even further down the page the user can **view followed authors.** Like the other sections this will be a link index, and like the favorites the user will have the ability to remove links by choosing to **stop following an author.**

Administration

Separate from the standard users of the site will be our administrators. These may be school faculty or possibly even other students. They will not be responsible for back-end management of the site (such as adding classes, assigning instructors to classes, or archiving past semesters). The primary difference for these users will be their home page on the site - they will have additional features.

Administration will have the ability to **review flags**. Notes that have been flagged (and not yet addressed) will be presented in a table at the top of their home page. They will have a link to the note in question, as well as the brief description that the user provided when flagging the note. When an administrator chooses to view a note, they will have the option to **edit a note**. This feature is intended to allow an admin to take out certain elements of a note that violate our note policy (but still leave the bulk of the note intact).

If an entire note violates our note policy, or there is not enough substance left in the note after all the editing, an admin may choose to **delete a note.** It is hoped that this will not need to be done often. However, if a user is a repeat offender, or violates our user policy, the administrator can also choose to **ban a user**. This will be done via their CSU email address.

External Design (User Interface):

login.php

The login.php page serves two functions to the Notezilla site. The first function it provides is the ability for an unknown user to login in and verify themselves. This is done using two text boxes which correlate to the user's name and password, which can uniquely identify the user in the database. The second function it provides is the ability for the new user to travel to the "Create an Account" page, where they can begin the process of creating a new account.

create_account.php

If a user is new to Notezilla, they will be able to navigate themselves from the login.php page to the create_account.php page. In the create_account.php page the user creates an account by filling in the blank text boxes labeled: "First Name", "Last Name", "Email", "Password", "Re-enter Password", and then clicking "Submit". The create_account.php page is the only other page, besides login.php, that non-members (members being: students, faculty, admin) can visit.

accountvalidation.php

This is a simple page that the user will return to from the email that they received during the account creation process. It will be sent a validation code that was generated based off their username and a special code. We will generate that code again on this page and make sure that the two match. Once validated we will enable their account on the system and link them to the login page.

notes_submission.php

In the notes_submission.php page, faculty or students can submit notes to the database under a specified class. Note submissions are handled in two ways: a member can either enter in their notes using the Notezilla text editor, or they can upload a PDF version of their notes to the database. Once the notes are written or the PDF is selected, the user chooses the class that the notes apply to and clicks “Submit”.

policies.php

Policies.php provides Notezilla with a way to legally operate under the authority of the University. It protects its users and its founders from the legal ramifications of Notezilla's misuse. A link to the Student Judicial Affairs is provided here, along with a mission statement.

catalog.php

A full list of classes is stored in the database and presented to the member upon request when navigating to the catalog.php page. At the catalog.php page a full list of all classes offered *during the current semester* are presented. Once the semester has ended, the administrator will enter in a new list of classes being offered for display on the catalog.php page. When a user navigates to the catalog page they can click on a class and view all submitted notes for that class they clicked on. The catalog page, ultimately, provides a high level view of the notes, categorized by the class they were taken it.

admin_home.php

The administrator for the site has their own page. This is done to allow special privileges and duties of the administrator to take place. At admin_home.php the administrator can find users, so that they may ban them. User bans can also be removed here. The admin_home.php page is also where the administrator can review or delete recent notes posted to the site. The administrator may also review/edit/delete flagged notes here as well. The function of admin_home.php page is to give the administrator a space in which to operate outside of the normal student/faculty use cases.

faculty_home.php

The faculty that have user accounts on the Notezilla site are given a different page from the student members. This is done to handle special use cases that faculty may have from students. At this time faculty_home.php gives faculty the ability to close classes that they do not wish for Notezilla to participate in. Here at this page the faculty may also follow particular students and their notes. The faculty may also view their notes that they have marked as “Favorites”.

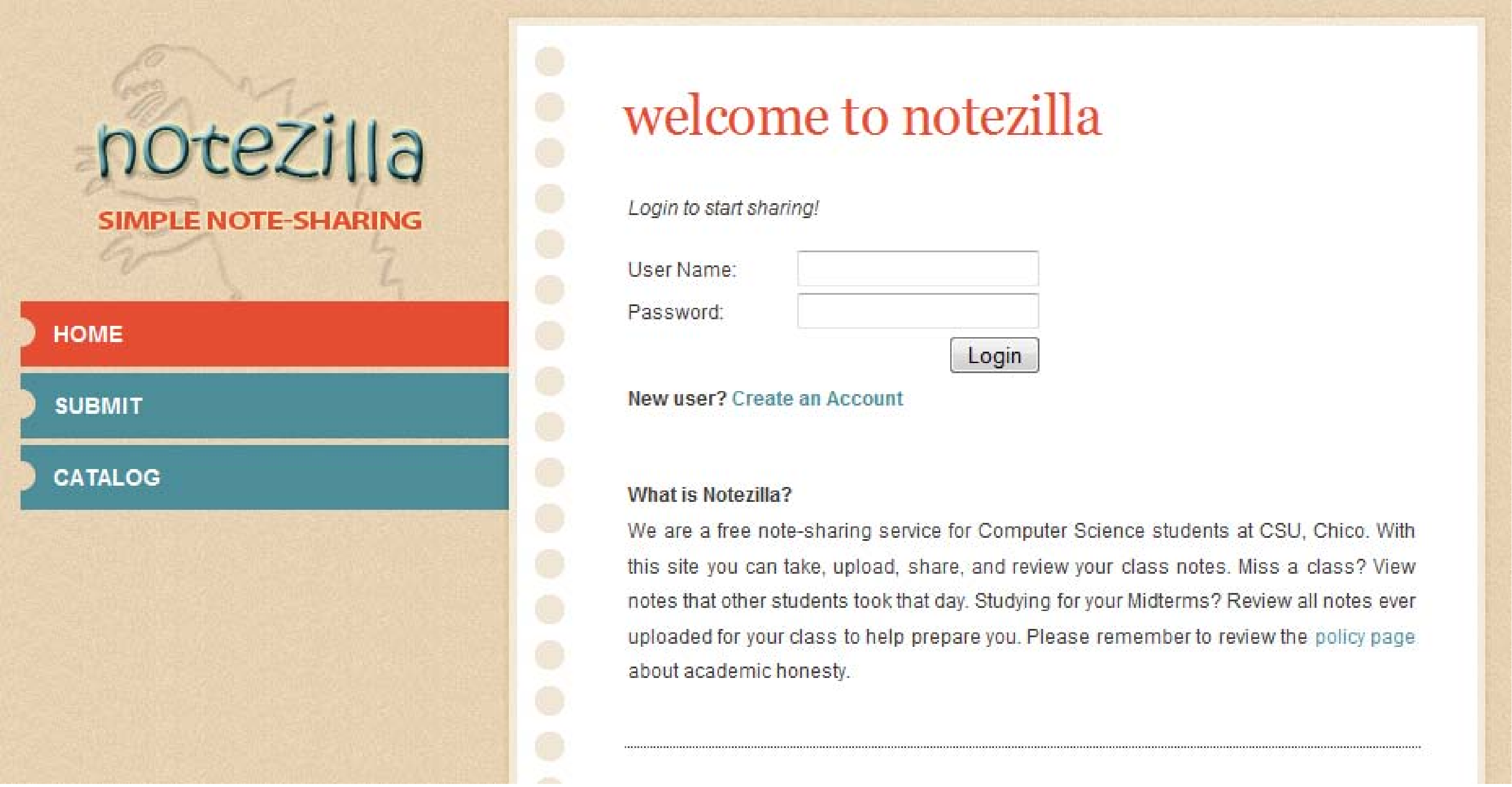
student_home.php

For the majority of the members on Notezilla, the student_home.php page will be the page viewed after they log in. This page allows, again, for the special use cases of the students to be presented. On the student_home.php page, students will be able to add/view notes on the classes that they are currently enrolled in. Students will also be able to view their own notes, on their home page. The student_home.php page also allows students the ability to view the notes of people who they are following, along with the notes they have marked as “Favorites”.

notes_retrieval.php

When a class link is visited, either from the catalog or homepage, the user is taken to the notes_retrieval.php page. At the retrieval page, members can see all the notes posted for a specific class. Once the user has viewed a note the user can mark the note as a “favorite”, the user can start “following” its author, or the user can flag the note as inappropriate (providing a reason if they choose). The notes_retrieval page allows for a specific class to be displayed via its notes.

index.php



welcome to notezilla

Login to start sharing!

User Name:

Password:

Login

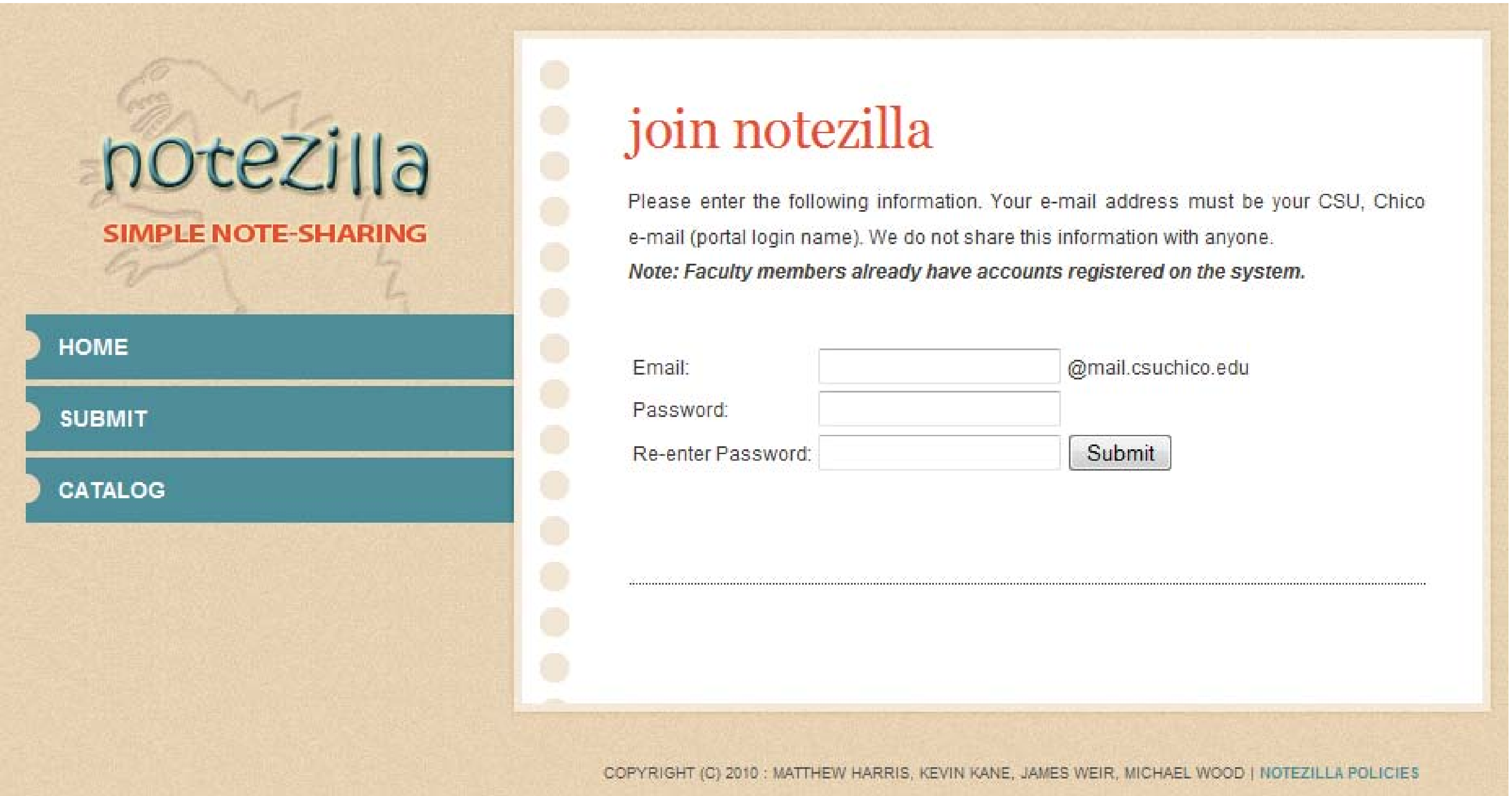
New user? Create an Account

What is Notezilla?

We are a free note-sharing service for Computer Science students at CSU, Chico. With this site you can take, upload, share, and review your class notes. Miss a class? View notes that other students took that day. Studying for your Midterms? Review all notes ever uploaded for your class to help prepare you. Please remember to review the policy page about academic honesty.

- Login Form (links to student_home.php / faculty_home.php / admin_home.php)
- Create Account button (links to create_account.php)
- About Us description

create_account.php



join notezilla

Please enter the following information. Your e-mail address must be your CSU, Chico e-mail (portal login name). We do not share this information with anyone.

Note: Faculty members already have accounts registered on the system.

Email: @mail.csuchico.edu

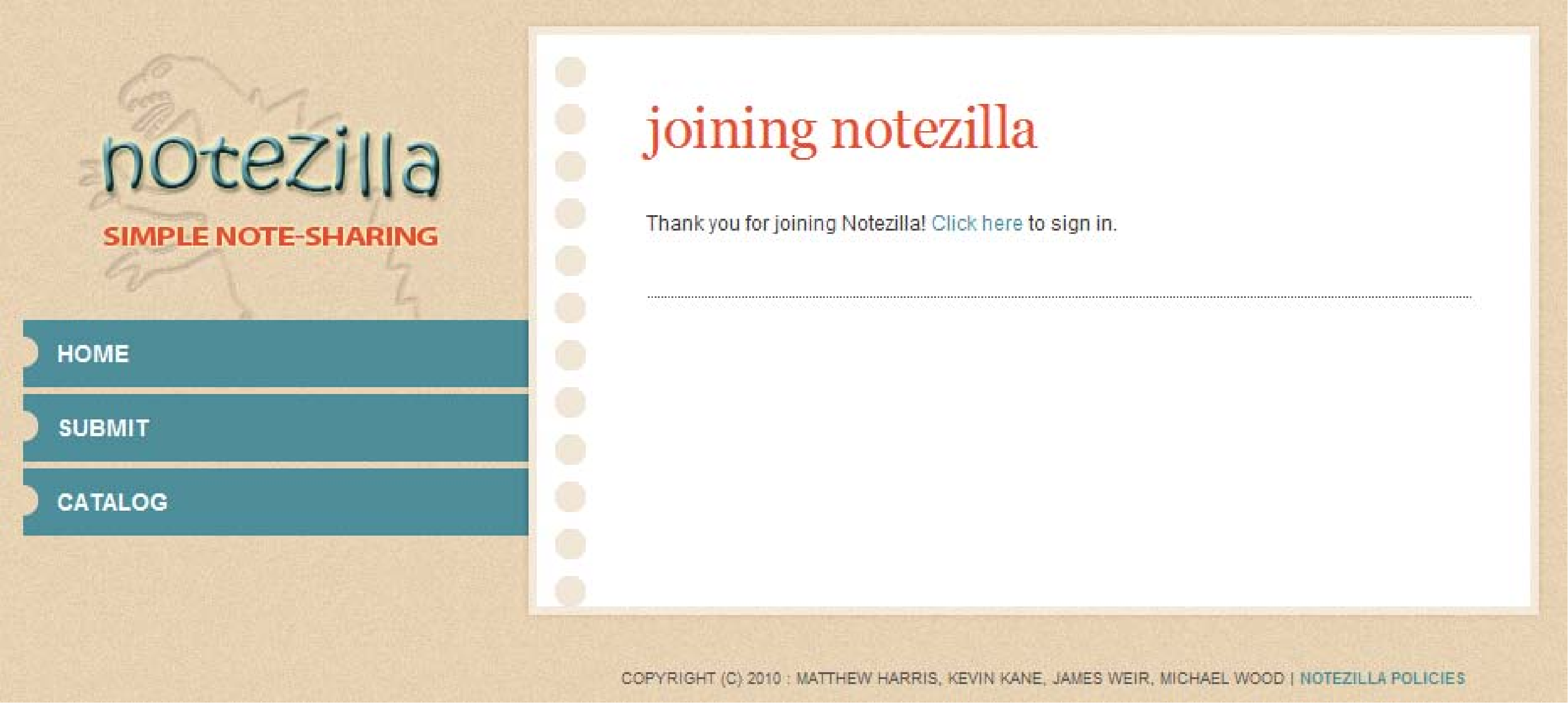
Password:

Re-enter Password:

COPYRIGHT (C) 2010 : MATTHEW HARRIS, KEVIN KANE, JAMES WEIR, MICHAEL WOOD | NOTEZILLA POLICIES

- User form (First Name, Last Name, Email, Password)

accountvalidation.php



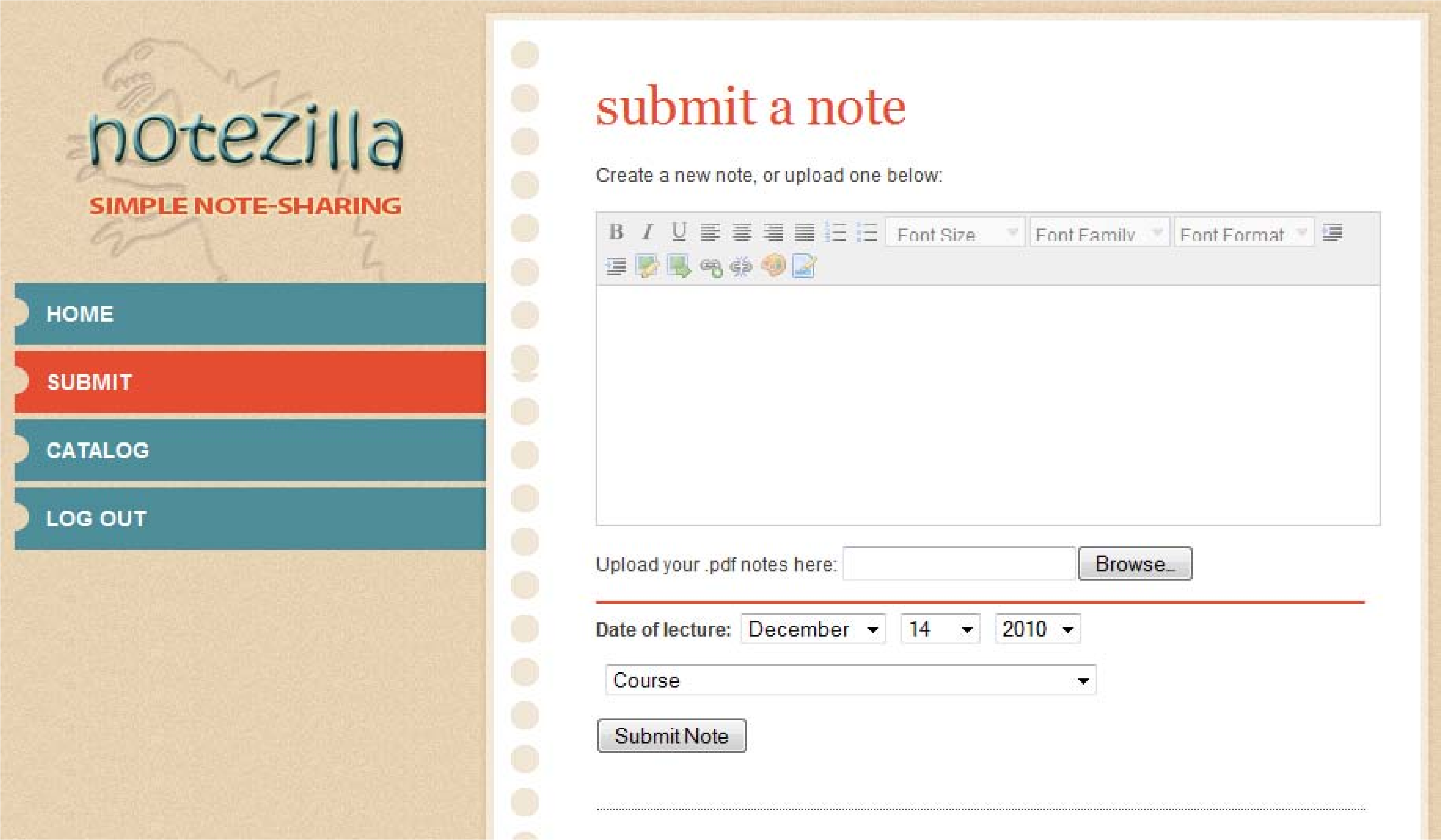
joining notezilla

Thank you for joining Notezilla! Click here to sign in.

COPYRIGHT (C) 2010 : MATTHEW HARRIS, KEVIN KANE, JAMES WEIR, MICHAEL WOOD | NOTEZILLA POLICIES

- Complete account creation process
- Provide user with login

note_submission.php



submit a note

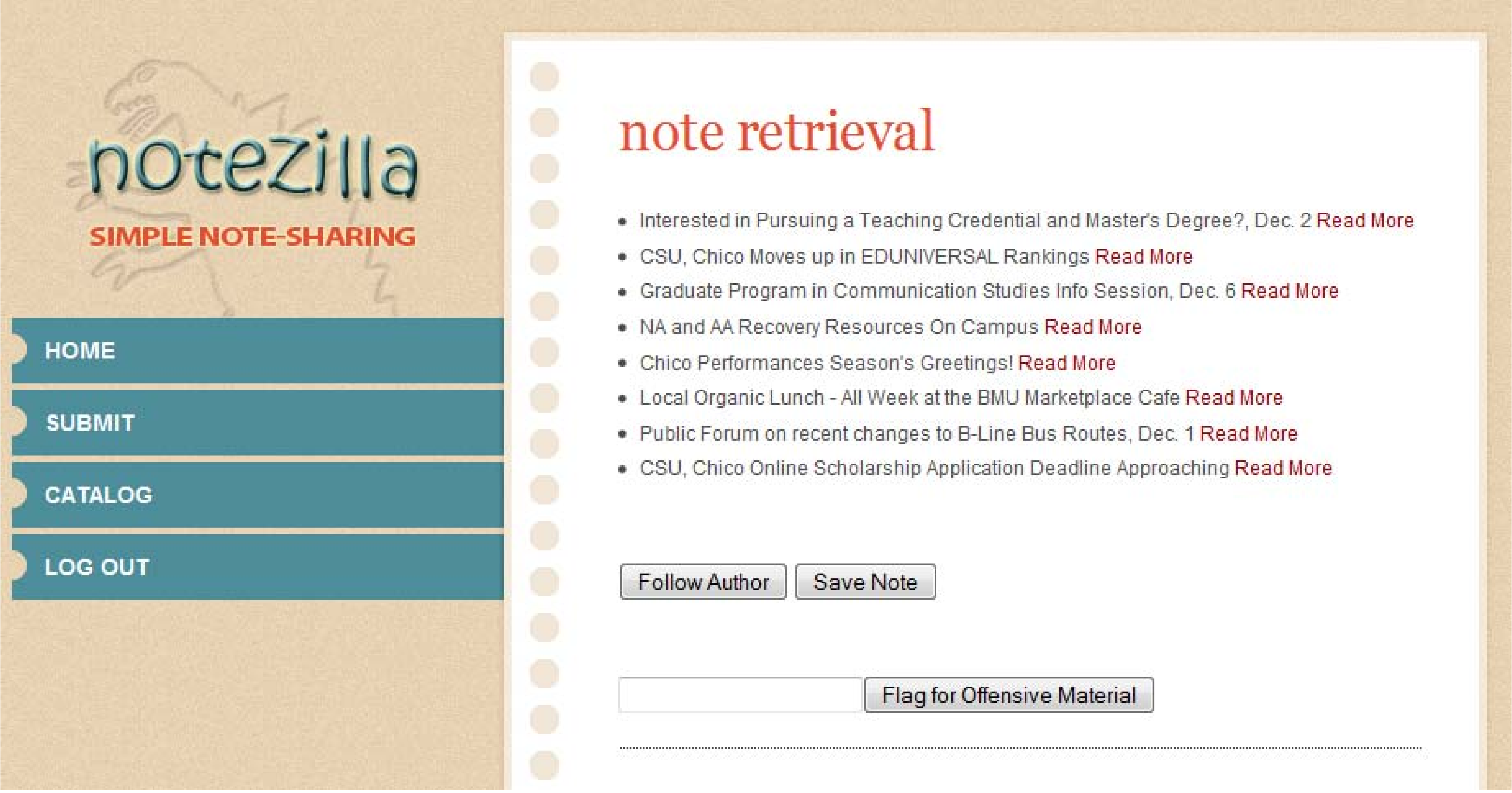
Create a new note, or upload one below:

Upload your .pdf notes here:

Date of lecture: December 14 2010

- Text Box for direct entry
- Submit PDF attachment

note_retrieval.php

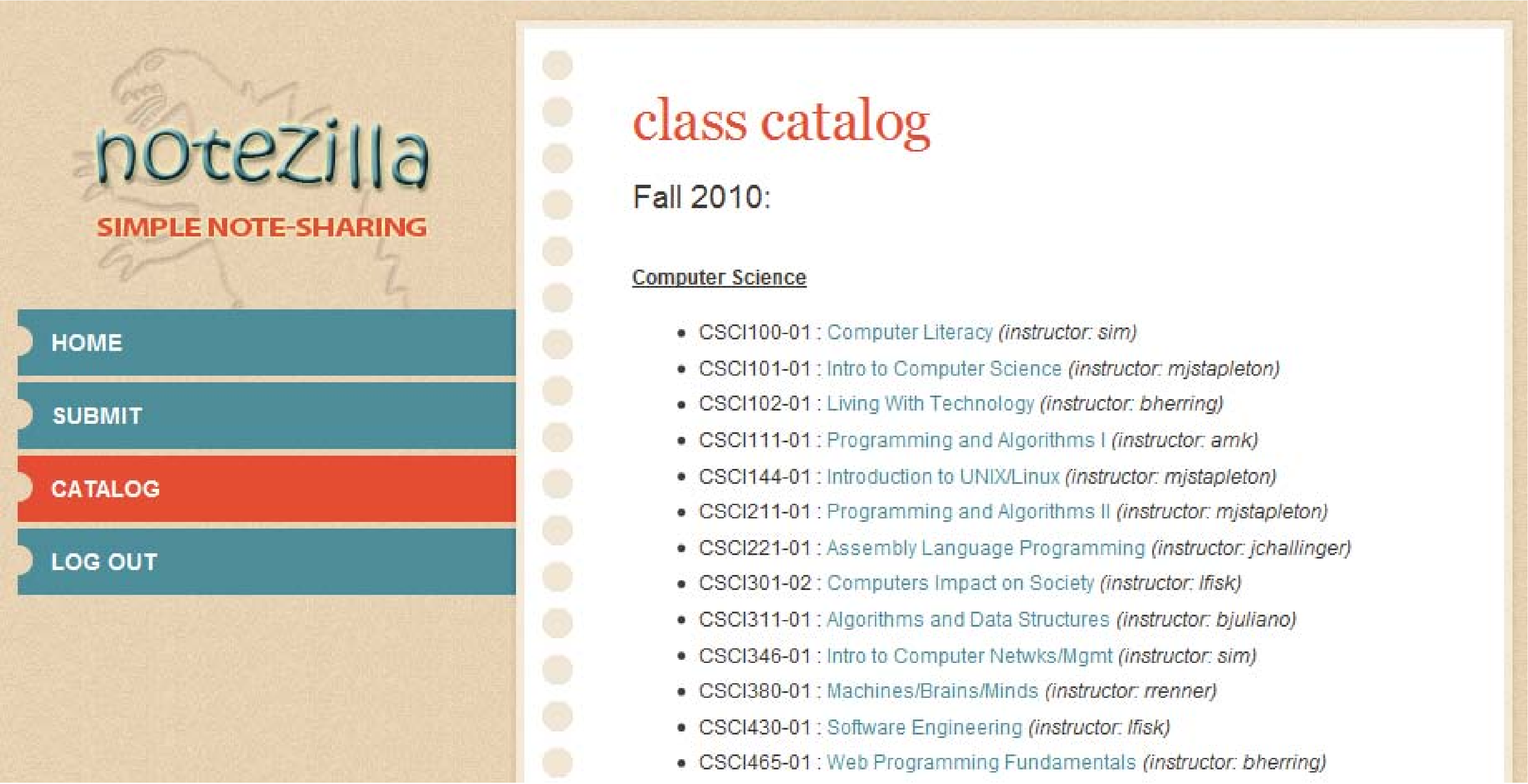


note retrieval

- Interested in Pursuing a Teaching Credential and Master's Degree?, Dec. 2 Read More
- CSU, Chico Moves up in EDUNIVERSAL Rankings Read More
- Graduate Program in Communication Studies Info Session, Dec. 6 Read More
- NA and AA Recovery Resources On Campus Read More
- Chico Performances Season's Greetings! Read More
- Local Organic Lunch - All Week at the BMU Marketplace Cafe Read More
- Public Forum on recent changes to B-Line Bus Routes, Dec. 1 Read More
- CSU, Chico Online Scholarship Application Deadline Approaching Read More

- Filter by author / date / date range
- Mark an author to follow
- Mark a note as a Favorite
- Flag a note as inappropriate

catalog.php



class catalog

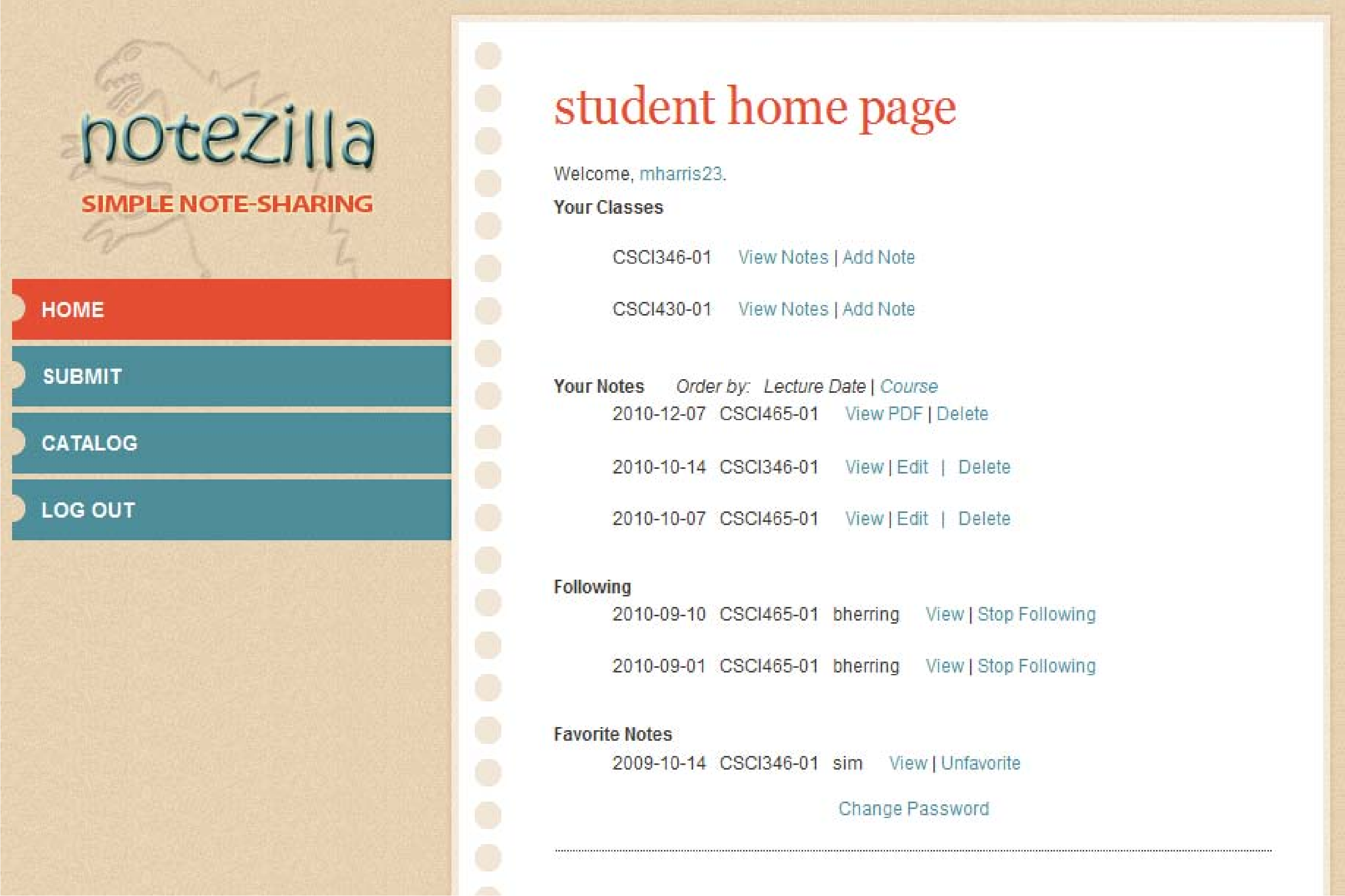
Fall 2010:

Computer Science

- CSCI100-01 : Computer Literacy *(instructor: sim)*
- CSCI101-01 : Intro to Computer Science *(instructor: mjstapleton)*
- CSCI102-01 : Living With Technology *(instructor: bherring)*
- CSCI111-01 : Programming and Algorithms I *(instructor: amk)*
- CSCI144-01 : Introduction to UNIX/Linux *(instructor: mjstapleton)*
- CSCI211-01 : Programming and Algorithms II *(instructor: mjstapleton)*
- CSCI221-01 : Assembly Language Programming *(instructor: jchallinger)*
- CSCI301-02 : Computers Impact on Society *(instructor: lfisk)*
- CSCI311-01 : Algorithms and Data Structures *(instructor: bjuliano)*
- CSCI346-01 : Intro to Computer Netwks/Mgmt *(instructor: sim)*
- CSCI380-01 : Machines/Brains/Minds *(instructor: rrenner)*
- CSCI430-01 : Software Engineering *(instructor: lfisk)*
- CSCI465-01 : Web Programming Fundamentals *(instructor: bherring)*

- All CSCI classes with links to their (notes_retrieval.php)

student_home.php



student home page

Welcome, mharris23.

Your Classes

CSCI346-01 View Notes | Add Note

CSCI430-01 View Notes | Add Note

Your Notes *Order by: Lecture Date | Course*

2010-12-07 CSCI465-01 View PDF | Delete

2010-10-14 CSCI346-01 View | Edit | Delete

2010-10-07 CSCI465-01 View | Edit | Delete

Following

2010-09-10 CSCI465-01 bherring View | Stop Following

2010-09-01 CSCI465-01 bherring View | Stop Following

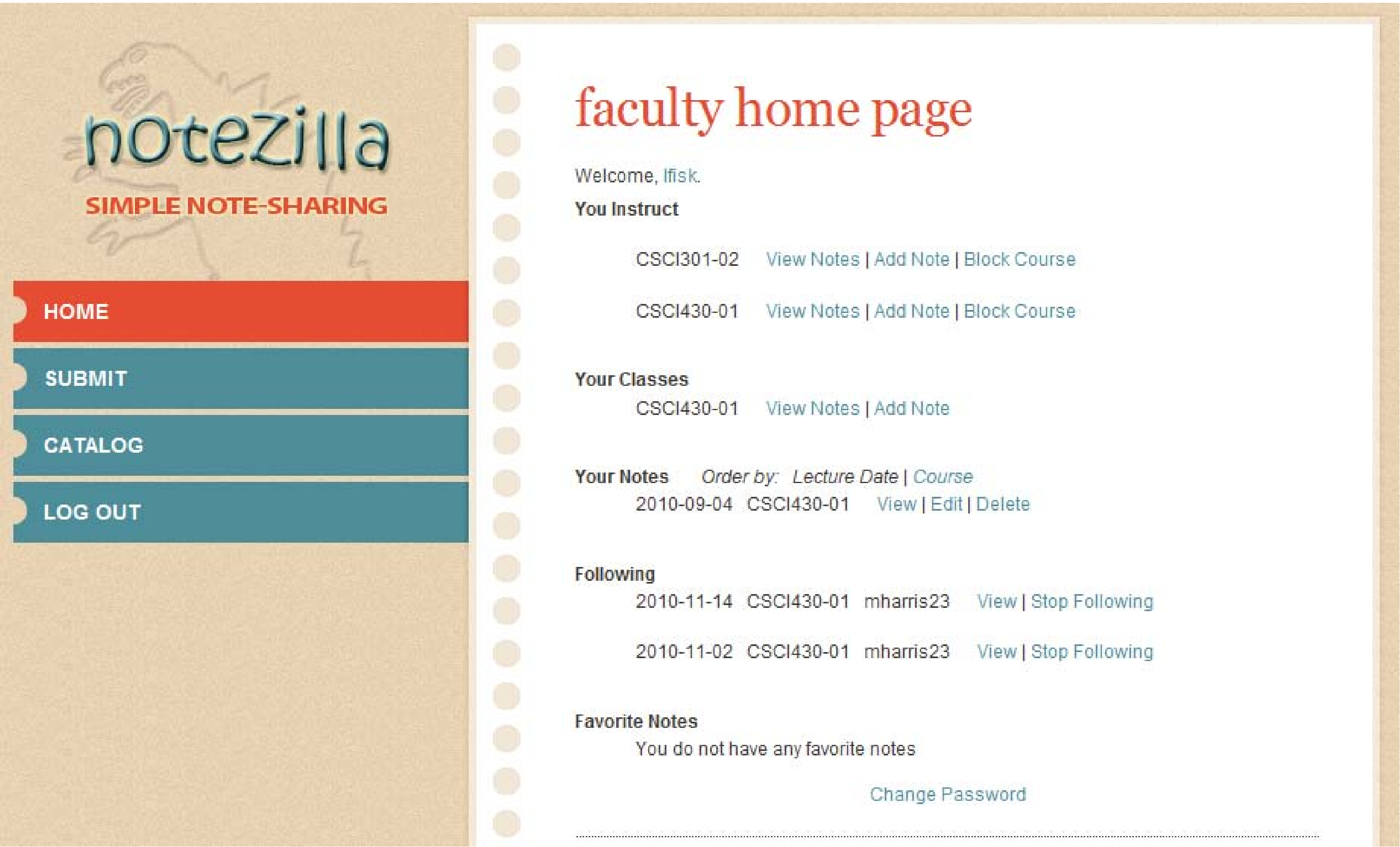
Favorite Notes

2009-10-14 CSCI346-01 sim View | Unfavorite

Change Password

- Links to view/edit/delete your notes
- Links to view/remove/report notes by followed authors
- Links to view/remove/report your favorite notes

faculty_home.php



faculty home page

Welcome, lfisk.

You Instruct

CSCI301-02 View Notes | Add Note | Block Course

CSCI430-01 View Notes | Add Note | Block Course

Your Classes

CSCI430-01 View Notes | Add Note

Your Notes *Order by: Lecture Date | Course*

2010-09-04 CSCI430-01 View | Edit | Delete

Following

2010-11-14 CSCI430-01 mharris23 View | Stop Following

2010-11-02 CSCI430-01 mharris23 View | Stop Following

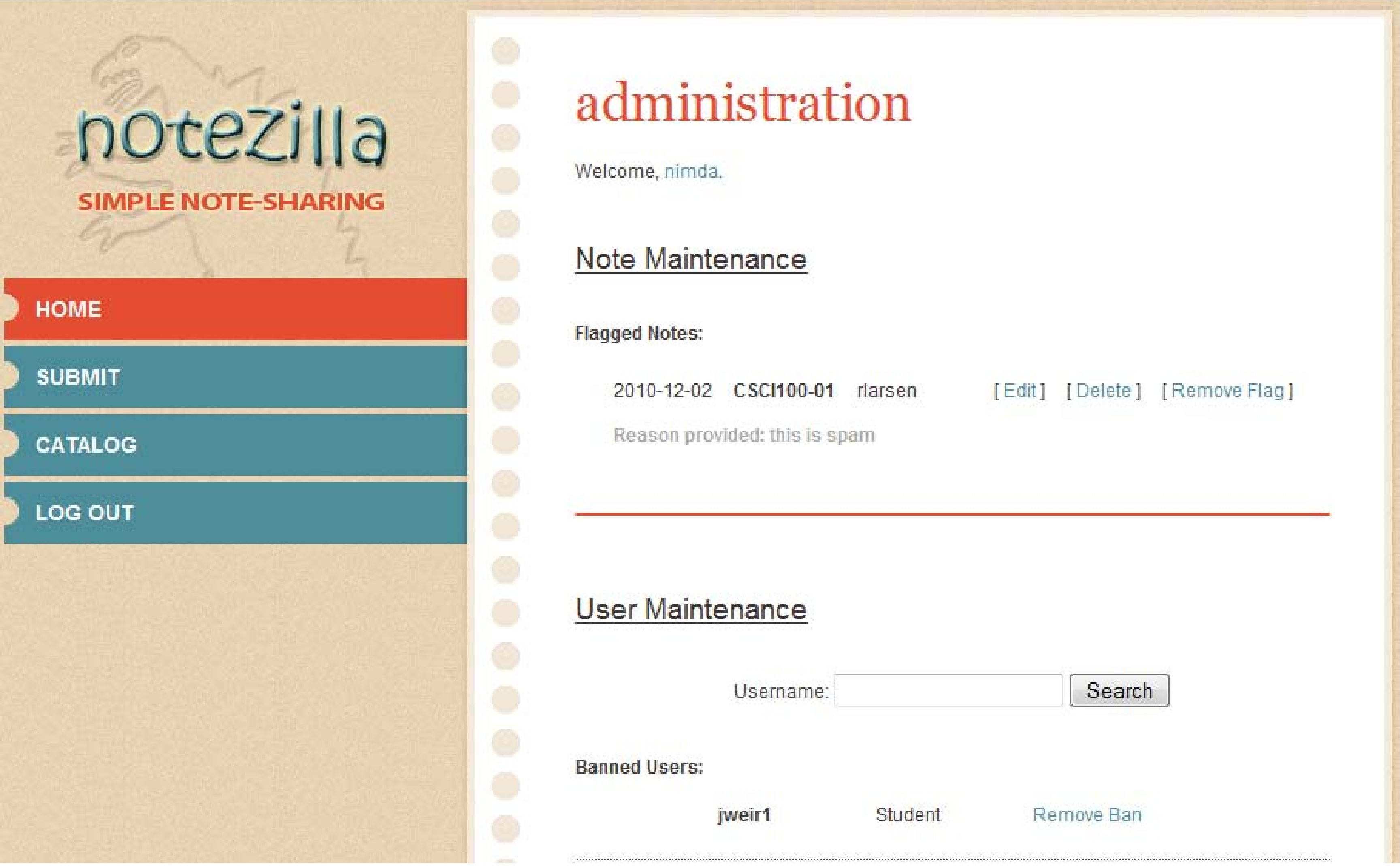
Favorite Notes

You do not have any favorite notes

Change Password

- Same features as student homepage.
- Block a class for uploading (opt-out program)

admin_home.php



administration

Welcome, nimda.

Note Maintenance

Flagged Notes:

2010-12-02 CSCI100-01 rlarsen [Edit] [Delete] [Remove Flag]

Reason provided: this is spam

User Maintenance

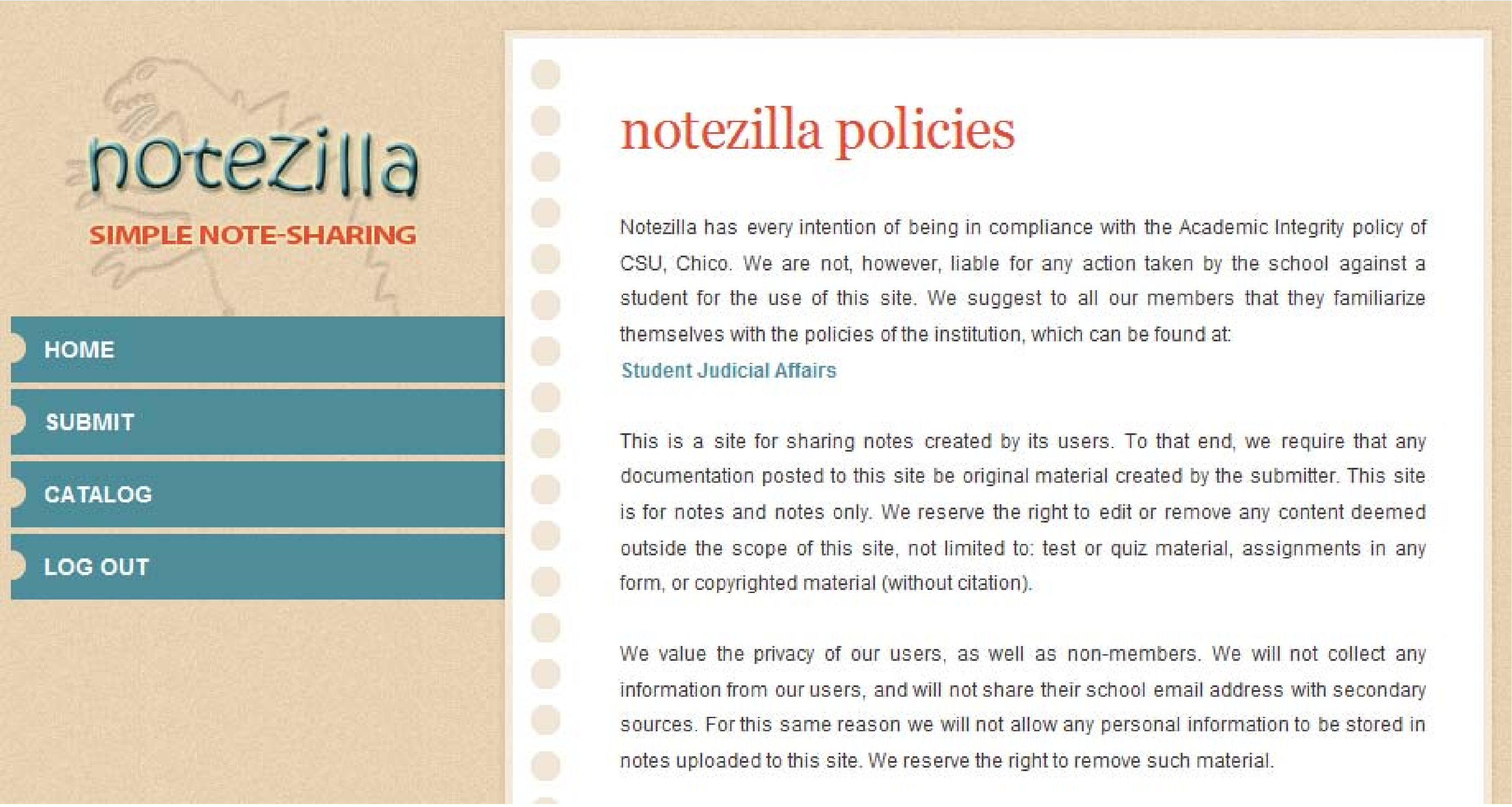
Username: Search

Banned Users:

jweir1 Student Remove Ban

- Review flagged notes
- Delete / Edit notes
- Ban users

policies.php



notezilla policies

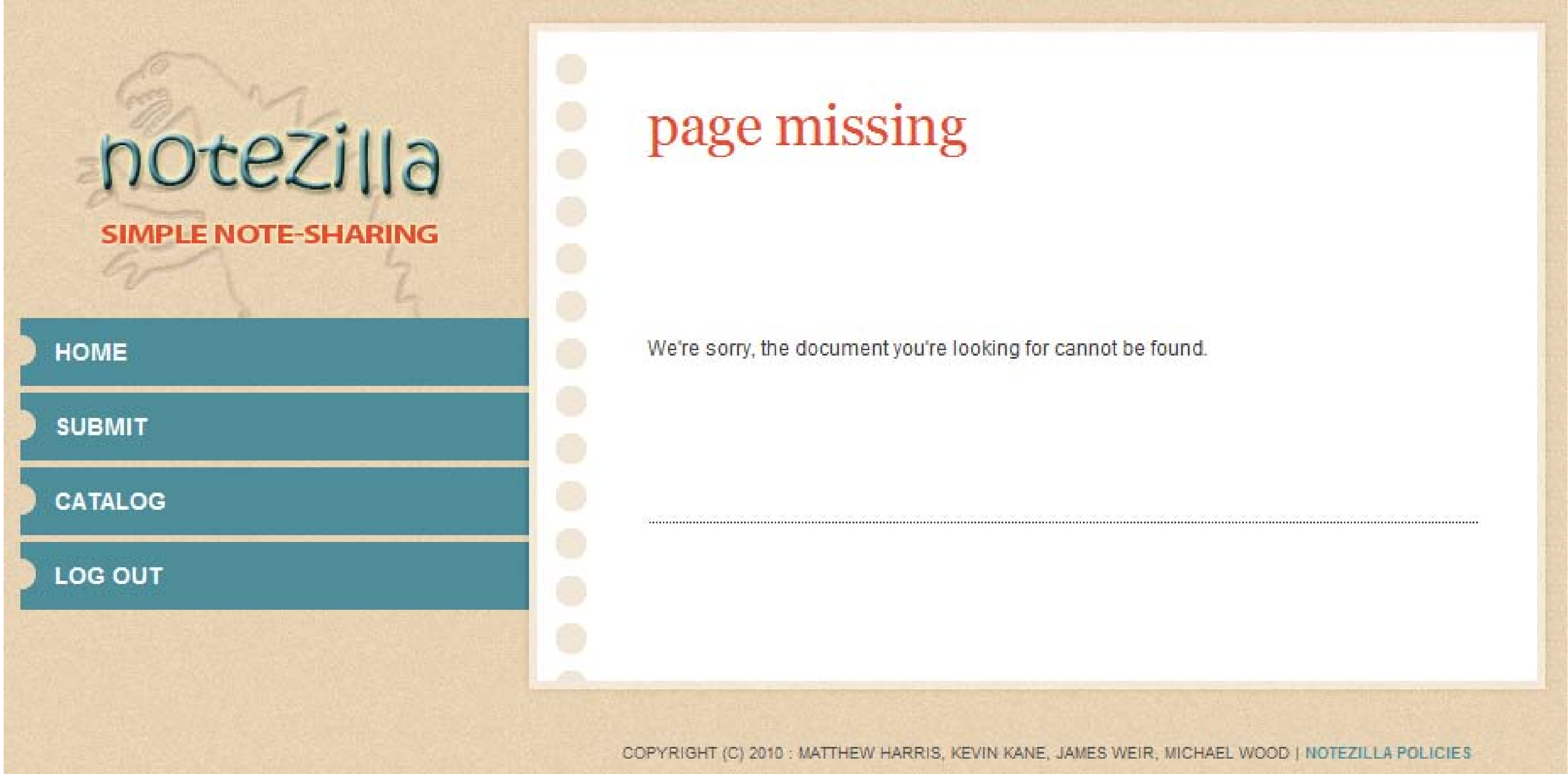
Notezilla has every intention of being in compliance with the Academic Integrity policy of CSU, Chico. We are not, however, liable for any action taken by the school against a student for the use of this site. We suggest to all our members that they familiarize themselves with the policies of the institution, which can be found at:
Student Judicial Affairs

This is a site for sharing notes created by its users. To that end, we require that any documentation posted to this site be original material created by the submitter. This site is for notes and notes only. We reserve the right to edit or remove any content deemed outside the scope of this site, not limited to: test or quiz material, assignments in any form, or copyrighted material (without citation).

We value the privacy of our users, as well as non-members. We will not collect any information from our users, and will not share their school email address with secondary sources. For this same reason we will not allow any personal information to be stored in notes uploaded to this site. We reserve the right to remove such material.

- All the rules and policies
- Policy links

error.php



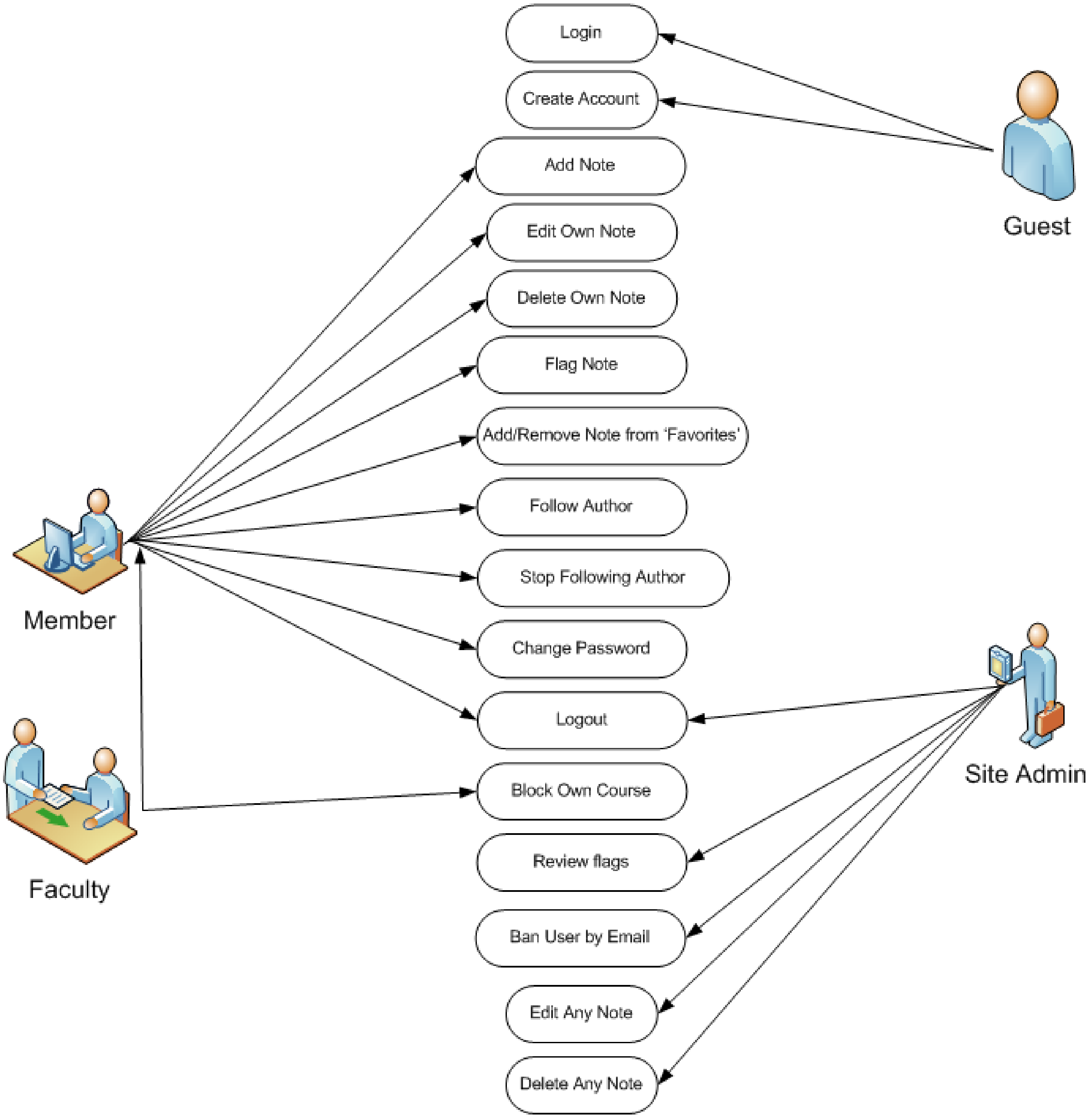
page missing

We're sorry, the document you're looking for cannot be found.

COPYRIGHT (C) 2010 : MATTHEW HARRIS, KEVIN KANE, JAMES WEIR, MICHAEL WOOD | NOTEZILLA POLICIES

- Provide themed pages for server errors (403 / 404)

Notezilla Use Case Diagram



Use Case Narrative:

Case #1: Login

Actor: Guest

Description: The Guest may enter in login credentials to get their role elevated to Member or Site Administrator

Case #2: Create Account

Actor: Guest

Description: The Guest visits website and selects "Create Account" then they enter in a user name, password and a unique email address. The Guest is then sent an email which allows them to verify their account and login as a Member.

Case #3: Add Note

Actors: Member, Faculty

Description: The Member selects 'SUBMIT' from the left panel then enters in text or uploads a PDF. After specifying the date and class, they may submit the note

Case #4: Edit Own Note

Actor: Member, Faculty

Description: From the student homepage, the Member selects Edit on the note they want to edit

Case #5: Delete Own Note

Actor: Member, Faculty

Description: From the student homepage, the Member selects Delete on the note they want to delete. The Member is prompted by a pop-up that asks if they want to delete their note. When they select "Yes" the Item is deleted from our database.

Case #6: Flag Note

Actor: Member, Faculty

Description: The Member may select 'Mark this Note as inappropriate' on any note to flag it

Case #7: Add/Remove Note from 'Favorites'

Actor: Member, Faculty

Description: Member selects 'Save This Note' on any note to add it to their 'Favorites'. From the student homepage the Member may select 'Remove' to remove a note from their favorites.

Case #8: Follow Author

Actor: Member, Faculty

Description: Member selects 'Follow Author' on any note to follow all of the notes an author creates in that class.

Case #9: Stop Following Author

Actor: Member, Faculty

Description: From the student homepage the Member may select 'Remove' to stop following an Author.

Case #10: Change Password

Actor: Member, Faculty

Description: The Member clicks 'Edit Profile' on their profile page. They can then edit their password.

Case #11: Logout

Actors: Member, Site Admin, Faculty

Description: The User selects the 'Logout' button on the top right of any page to logout.

Case #12: Block Own Course

Actor: Faculty

Description: Faculty visits their course page and selects 'block own course', they are presented with a message asking them to confirm. If yes, the course is blocked and no new notes can be created in that section of the course

Case #13: Review Flags

Actor: Site Admin

Description: Site Admin selects 'Review Flags' from their profile page and is shown all notes that are flagged, the admin can then decide what to do with the flagged note.

Case #14: Ban User by Email

Actor: Site Admin

Description: Site Admin selects 'Ban User' from their profile page and enters the email of the user to ban.

Case #15: Edit Any Note

Actors: Site Admin

Description: The Admin can select 'Edit' on any note to make changes.

Case #16: Delete Any Note

Actor: Site Admin

Description: The Admin may select 'Delete' on any note to delete it.

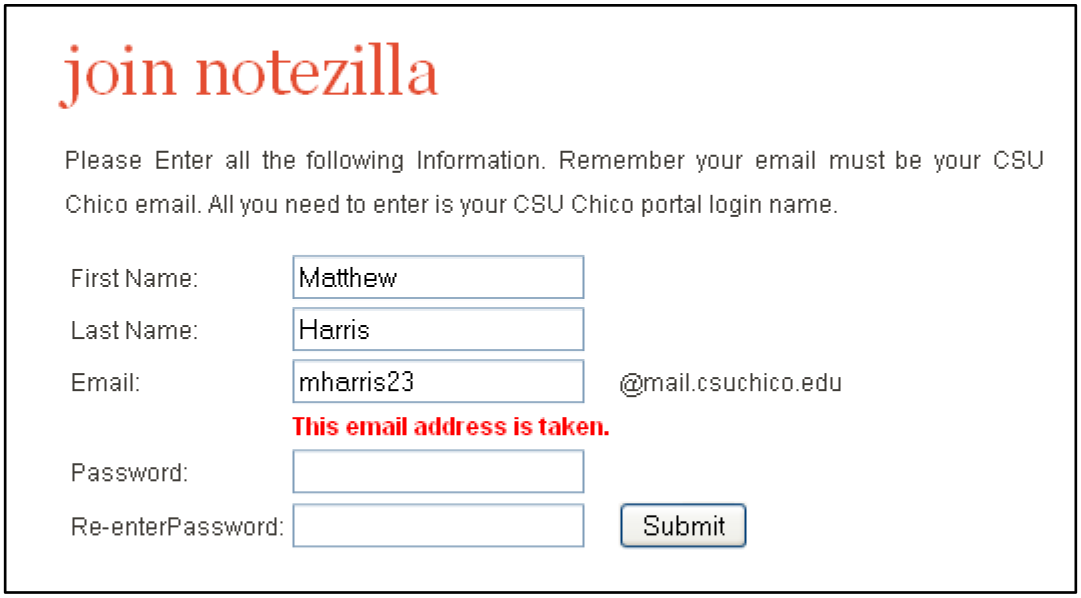
Data Validation:

Our only user input comes from the *Create Account*, *Login*, and *Submit Note* pages.

- When submitting notes we are going to enter a block of text. To prevent any SQL injections, we will use the PHP function mysql_real_escape_string().

- On the login page we will also look to prevent SQL injections and if a username is entered that is not in our system the site will return to the page, with a red message stating "there is no account under this id". If the username is valid but the password does not match, the page will return with a red error "Incorrect Password".

- The last page with user submitted data is the "create an account" page. Again we will check all input for SQL injections. We will check that names only contain letters and that the email (portal username) is only letters and numbers. If any input is invalid the page will return with a red error stating which input was incorrect. See the following image for an example:



join notezilla

Please Enter all the following Information. Remember your email must be your CSU Chico email. All you need to enter is your CSU Chico portal login name.

First Name:	Matthew	
Last Name:	Harris	
Email:	mharris23	@mail.csuchico.edu
	This email address is taken.	
Password:		
Re-enterPassword:		

Future Features:

The following is a list of features we would like to implement in future builds of the site. These go beyond our first completed version. Some aspects of these features may be planned for in our initial schema, but many will not.

- Expansion to other departments in CSU Chico, and eventually other schools
- Store a student's enrolled classes (regardless of whether they have notes in them)
- Allow importing from other note systems (Evernote, Microsoft OneNote, etc)
- Search ability for notes stored in non-image formats
- User-provided tags to search & sort by content categories
- User-provided headers (title, brief description, etc)
- Archiving of notes from prior semesters (automated if possible)
- A more automated process for linking faculty with their classes
- Voting system for notes (to filter the best to the top)
- Alternate "printer-friendly" display for notes
- Viewer controls (font/size/color) for accessibility to handicapped users
- Download options (compressed ZIP archives for an entire class, etc)

Internal Design (Modules):

admin_home.php

Purpose: To provide specific use cases to administrators.

Functionality: If session and administrator, allow to view site, else redirect to index. Provide administrator the ability to view recent entries, edit recent entries, delete recent entries. Allow administrator to view comments on flagged

catalog.php

Purpose: To provide a list of all currently active classes.

Functionality: If session, allow to view module, else redirect to index. Provide session users the ability to view notes based on a specific class.

create_account.php

Purpose: To allow users the ability to create student accounts.

Functionality: No session required. Allow new users the ability to create a class by entering in user information, email validation required for database entry to succeed.

accountvalidation.php

Purpose: To activate the user in the database.

Functionality: No session required. When the user receives their email validation it will link back here with an encrypted hash. This page will verify that their hash matches what it should and enables their account.

faculty_home.php

Purpose: To provide specific use cases to faculty.

Functionality: If session and faculty, allow to view site, else redirect to index. Provide faculty the ability to view current classes. Within those current classes, allow faculty to view notes for specific class, add note for specific class, and close a specific class. Allow faculty the ability to see recent notes added by writers they are following, of which they can edit or delete. Give faculty the ability to see "favorite" notes, of which they can edit or delete.

index.php

Purpose: To allow users a place to log in, and link to creating a new page.

Functionality: No session required to view page. Allow users the ability to authenticate themselves using portal name and password. Once authentication returns true, give users a session and redirect to appropriate home (faculty, student, or admin).

notes_retrieval.php

Purpose: To allow users the ability to view specifically selected notes.

Functionality: Session required, else redirect to index. Allow users the ability to see all submitted notes for a specific class. Allow users to rearrange or filter the notes by authors, date, or a specific date range. Allow users to mark note as inappropriate, save an author to "follow", or add a note to "favorites".

notes_submission.php

Purpose: To allow users the ability to submit notes into the database

Functionality: Session required, else redirect to index. Allow users to submit a note either via text box or via PDF. If the user submits via PDF, allow user to browse their computer for file to input. Validate that file is a MIME-type 'application/pdf'. Submitted files are placed into the database.

student_home.php

Purpose: To provide specific use cases to students.

Functionality: Session required, else redirect to index. Allow users to submit or view notes for currently enrolled classes. Allow users to view, edit, or delete their own notes. Allow users to view, remove or report, notes that they are "following". Allow users to view, remove, or report, their favorite notes.

db.php

Purpose: An object class, other modules the ability to log into the database without having to write it each time.

Functionality: Allow other modules to log into the database and use it.

authentication.php

Purpose: An object class giving other modules the ability to authenticate users.

Functionality: Receive name and password, return true if user is in database and password is correct, else return false.

session.php

Purpose: An object class giving other modules the ability to know if a user has already been authenticated and is in a "session"

Functionality: Allow other modules to know if the user has already been authenticated.

header.php

Purpose: A header file for the HTML content to include

Functionality: This page includes all the META tags, including a title function that will determine the correct title to display on the browser (based on the URI).

footer.php

Purpose: A footer file for the HTML content to include

Functionality: This page includes the copyright information, the termination of the page layout, and a link to our policies page.

functions.php

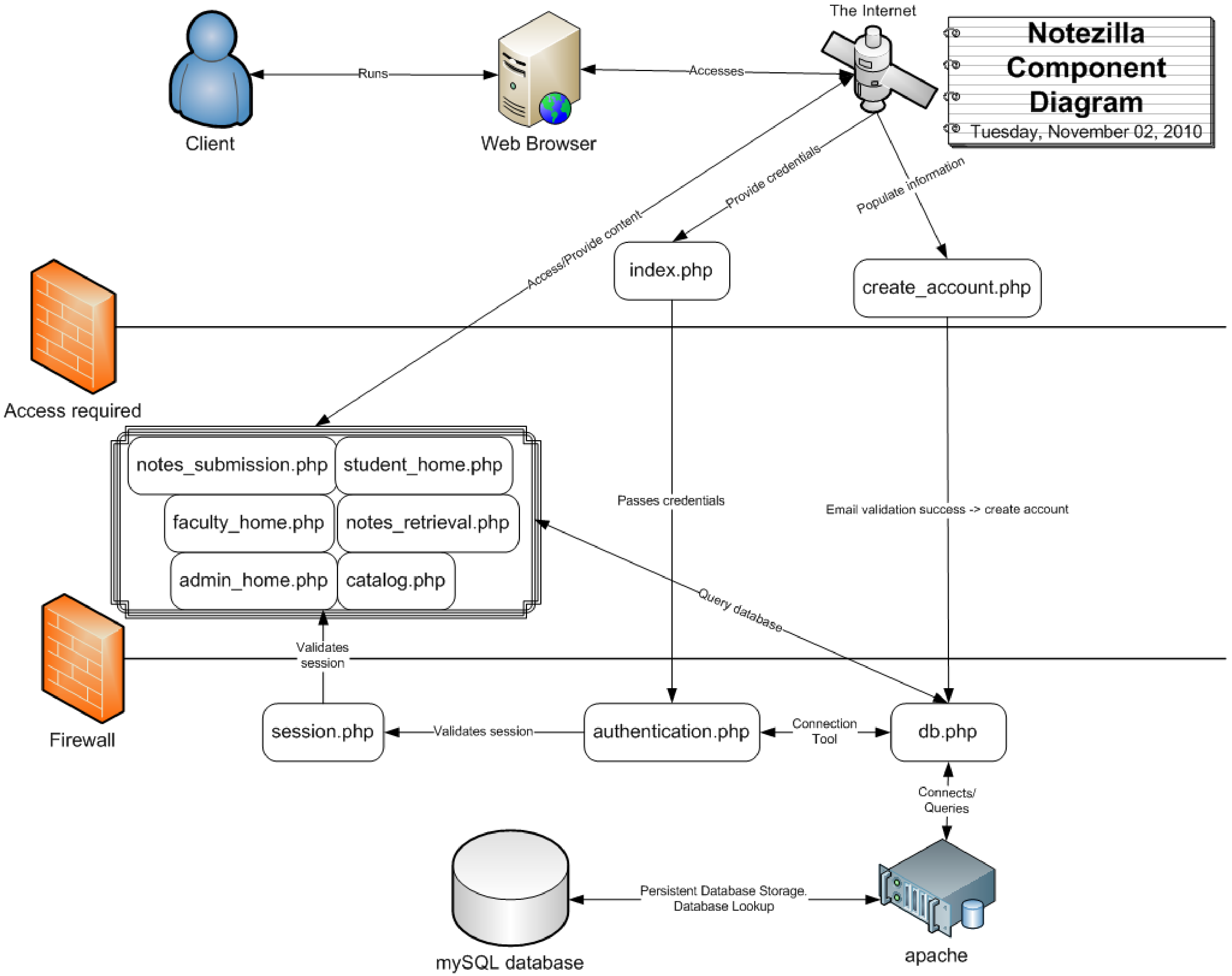
Purpose: Sidebar navigation for the site

Functionality: This page will be included along with the header/footer and will contain all the navigation links on the site. This way we can ensure every page has the same look. The Log Out function will appear here.

error.php

Purpose: Provide friendlier HTTP error messages

Functionality: This is a simple skin for the default Apache 403/404 error messages. It will allow them to stay within the style of the site, and hide private server information from the general public.


Notezilla
Component
Diagram
Tuesday, November 02, 2010
Client
Runs
Web Browser
Accesses
The Internet
Access/Provide content
Provide credentials
Populate information
index.php
create_account.php
Access required
notes_submission.php
student_home.php
faculty_home.php
notes_retrieval.php
admin_home.php
catalog.php
Passes credentials
Email validation success -> create account
Query database
Validates session
Firewall
session.php
Validates session
authentication.php
Connection Tool
db.php
Connects/ Queries
Persistent Database Storage. Database Lookup
mySQL database
apache

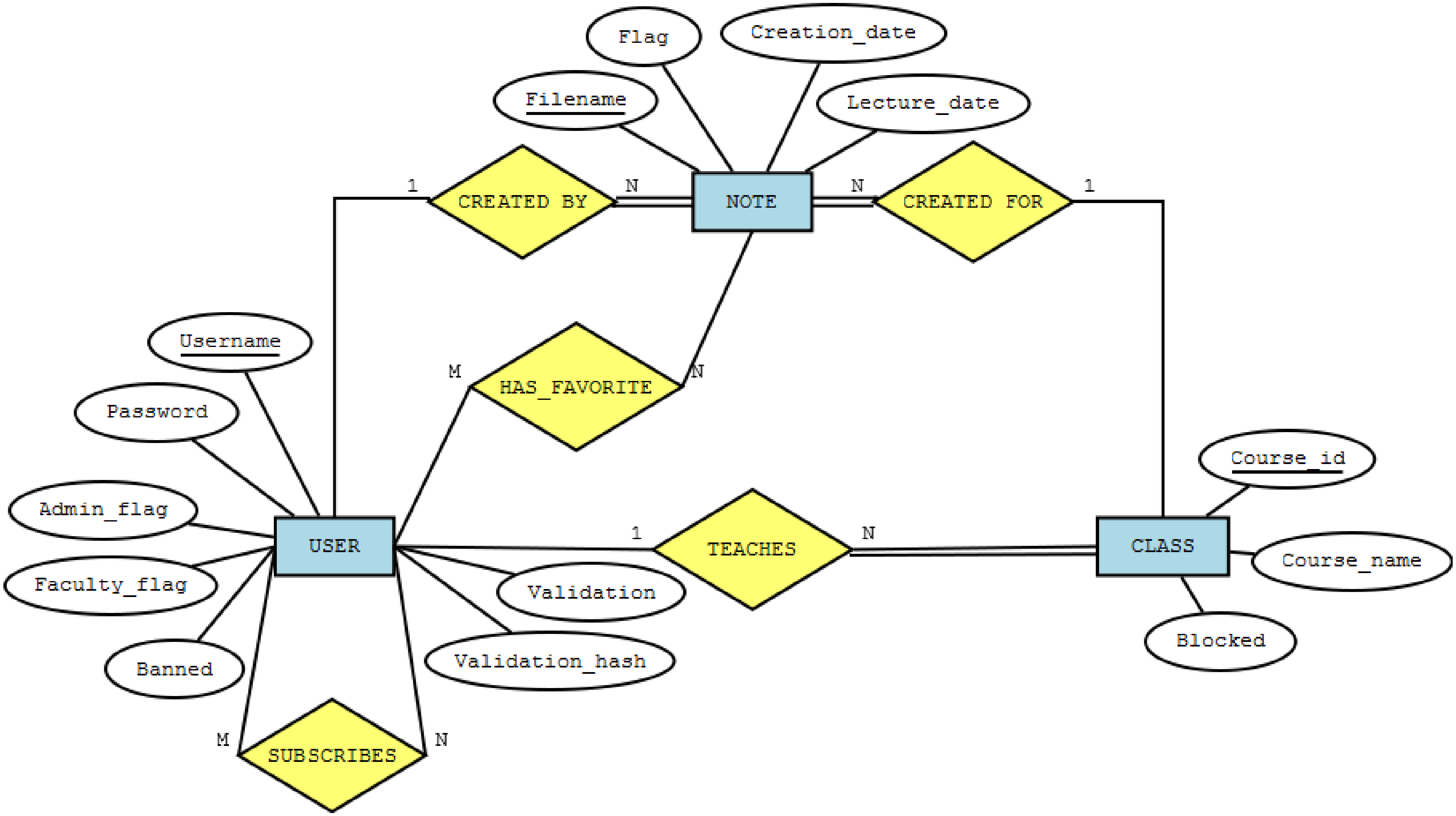
Task List:

Module	**Developer**	**Total Hours**	**Hours Spent**	**Complete**
student_home	Kevin	12	15	100%
faculty_home	Kevin	12	15	100%
misc	Kevin	6	5	100%
index	Michael	4	15	100%
catalog	Michael	4	5	100%
notes_retrieval	Michael	22	21	100%
notes_submission	Matt	20	30	100%
create_account	Matt	10	15	100%
admin_home	James	10	20	100%
auth/session	James	4	4	100%
db connect	James	3	3	100%
backend setup	James	13	15	100%
Project Total:		**120**	**163**	**100%**

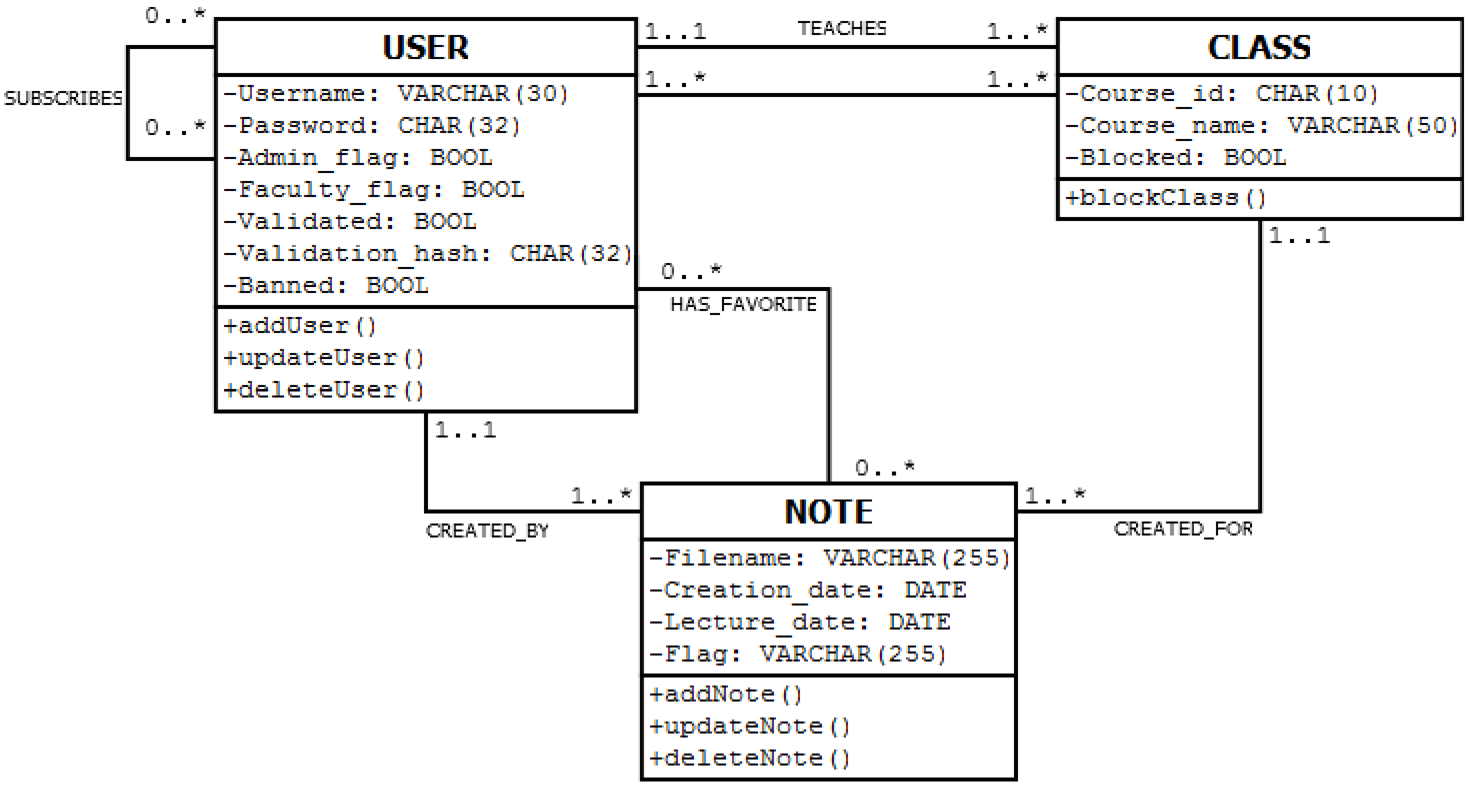
Developer	**Total Hours**	**Hours Spent**	**Hours Left**	**Complete**
Kevin	30	35	0	100%
Michael	30	41	0	100%
Matt	30	45	0	100%
James	30	42	0	100%

Entity Relationship Diagram:

We start our schema model section with an ER diagram. We can see from this that we will need four tables in our database. We can see from this that NOTE requires total participation from USER and CLASS (meaning a note cannot exist without an author & class). Most of our relationships will exist as a single column in a table (for example: CLASS will have an Instructor attribute that will be a foreign key to USER).

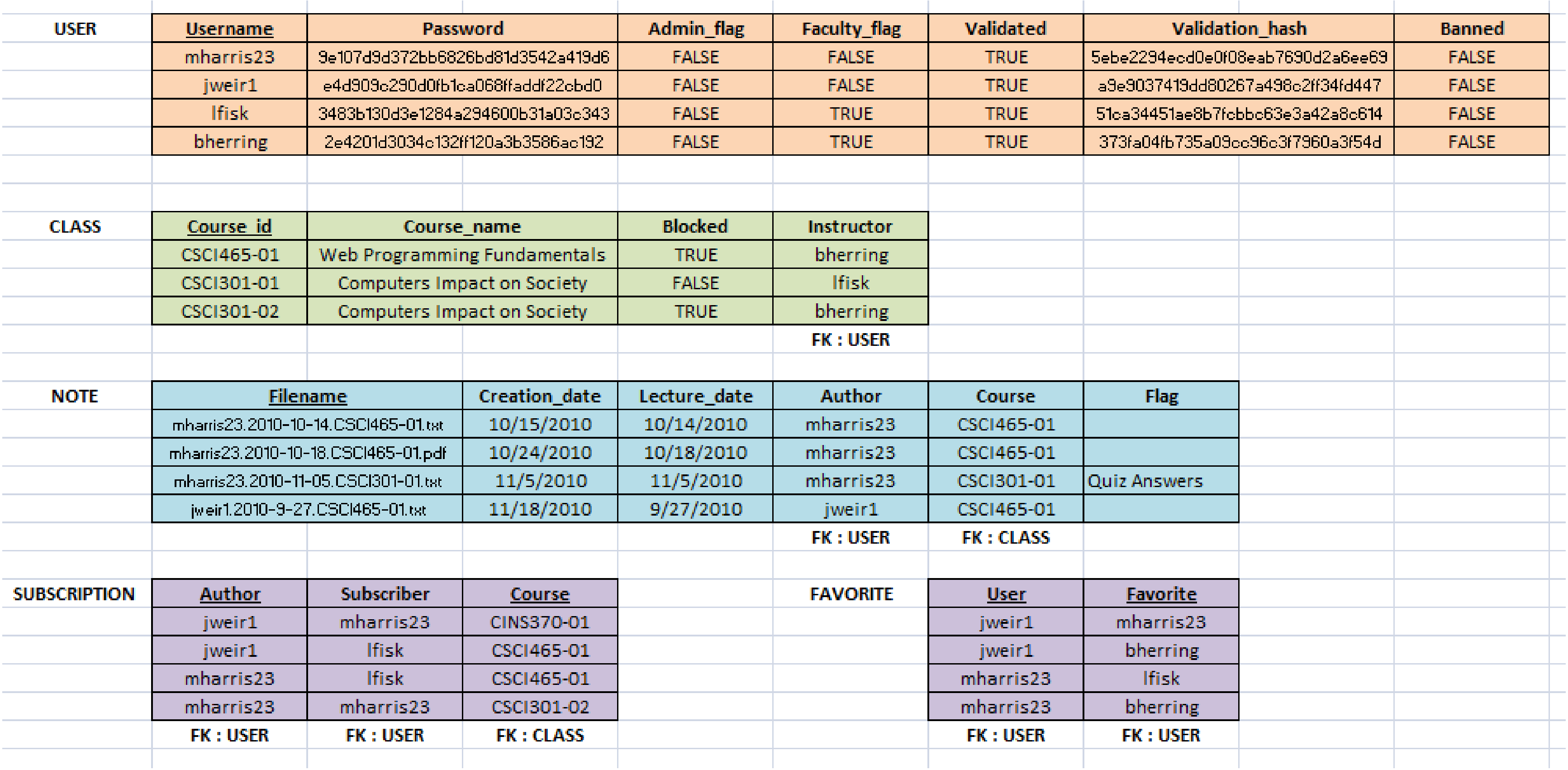


Object Model:



Our object model shows the data types we plan to use. These data types are specifically for MySQL. We also included some planned functions for each of our classes. The USER's password field will be an MD5 hash (stored as a 32-bit value). Note filenames will be stored as relative paths (not absolute). Lastly, we will be using Boolean flags to keep track of administrators and blocked classes.

Entity Relationship Mapping (with sample data):



USER

Username	Password	Admin_flag	Faculty_flag	Validated	Validation_hash	Banned
mharris23	9e107d9d372bb6826bd81d3542a419d6	FALSE	FALSE	TRUE	5ebe2294ecd0e0f08eab7690d2a6ee69	FALSE
jweir1	e4d909c290d0fb1ca068ffaddf22cbd0	FALSE	FALSE	TRUE	a9e9037419dd80267a498c2ff34fd447	FALSE
lfisk	3483b130d3e1284a294600b31a03c343	FALSE	TRUE	TRUE	51ca34451ae8b7fcbbc63e3a42a8c614	FALSE
bherring	2e4201d3034c132ff120a3b3586ac192	FALSE	TRUE	TRUE	373fa04fb735a09cc96c3f7960a3f54d	FALSE

CLASS

Course_id	Course_name	Blocked	Instructor
CSCI465-01	Web Programming Fundamentals	TRUE	bherring
CSCI301-01	Computers Impact on Society	FALSE	lfisk
CSCI301-02	Computers Impact on Society	TRUE	bherring
			FK : USER

NOTE

Filename	Creation_date	Lecture_date	Author	Course	Flag
mharris23.2010-10-14.CSCI465-01.txt	10/15/2010	10/14/2010	mharris23	CSCI465-01	
mharris23.2010-10-18.CSCI465-01.pdf	10/24/2010	10/18/2010	mharris23	CSCI465-01	
mharris23.2010-11-05.CSCI301-01.txt	11/5/2010	11/5/2010	mharris23	CSCI301-01	Quiz Answers
jweir1.2010-9-27.CSCI465-01.txt	11/18/2010	9/27/2010	jweir1	CSCI465-01	
			FK : USER	FK : CLASS	

SUBSCRIPTION

Author	Subscriber	Course
jweir1	mharris23	CINS370-01
jweir1	lfisk	CSCI465-01
mharris23	lfisk	CSCI465-01
mharris23	mharris23	CSCI301-02
FK : USER	FK : USER	FK : CLASS

FAVORITE

User	Favorite
jweir1	mharris23
jweir1	bherring
mharris23	lfisk
mharris23	bherring
FK : USER	FK : USER

This ER Map helps to show the relation our foreign keys will play (linking the tables), and gives us some sample data. This is a snapshot of what the database might look like down the road after some use.

Technical Issues

- Had issues with the Jaguar server and moved off-site to Linode (email verification, innoDB tables)

Revision History

- Milestone 4.0
  - Task List
  - Entity Relationship Diagram
  - Object Model
  - ER Map

- Milestone 5.0
  - Requirements
  - External Design
  - Internal Design
  - Task List
  - Entity Relationship Diagram
  - Object Model
  - ER Map
  - *Not updated for 5.0: Page Views (mock-ups)*

- Milestone 6.0
  - Internal Design
  - Task List

- Milestone 7.0
  - Page Views (mock-ups)
  - Task List