

Tufts University

School of Engineering
Department of Electrical and Computer Engineering



Comp 97: Design Document

Fall 2013

Name:

Jillian Silver
Josh Fishbein
Jillian.Silver@tufts.edu
Joshua.fishbein@tufts.edu

Submitted to:

Ming Chow

Table of Contents

	Section	Page
I	Abstract	3
II	Application Specific Technologies	
3-4		
III	Database	4-5
	1. <i>Tables</i>	
	2. <i>Models</i>	
	3. <i>Database Relations</i>	
IV	Searching Component	5
V	Static Pages	6
VI	User Login	6
	1. <i>User Signup</i>	
	2. <i>User Login</i>	
VII	Project Upload	6-7
VIII	Views	7-9
IX	Controllers	9
X	Testing	
9-10		

I. Abstract

The mission of this project is to enhance on campus student projects (particularly those related to sustainability) by providing students with the means to view each other's work and learn from previous projects. This will be done by constructing a web application allowing users to upload and search for one another's projects. This will be hosted on a Tufts Linux server.

II. Application Specific Technologies

This application will be built on the Ruby On Rails framework and will use the following technologies:

- Rails version 4.0.0
- Ruby version 1.9.3
- RVM 1.23.14 (stable)

The following ruby gems will be used source

- rails 4.0.0
- bootstrap-sass 2.3.2.0
- bcrypt-ruby 3.0.1
- faker 1.1.2
- will_paginate 3.0.4
- bootstrap-will_paginate 0.0.9
- sass-rails 4.0.0
- uglifier 2.1.1
- coffee-rails 4.0.0
- jquery-rails 3.0.4
- turbolinks 1.1.1
- jbuilder 1.0.2

development gems

- mysql 2.9.1
- rspec-rails 2.13.1
- paperclip 2.7

test environment gems

- selenium-webdriver 2.35.1
- capybara 2.1.0
- factory_girl_rails 4.2.0
- cucumber-rails 1.4.0
- database_cleaner github: 'bmabey/database_cleaner'
- libnotify 0.8.0

production gems
- mysql 2.9.1
- rails_12factor 0.0.2
- paperclip 2.7

The web application will be deployed to a Tufts owned Linux server.

III. Database

The persistent data for this web application will be stored in a MySQL relational database.

1. Tables

This database will have a User, a Project, and a Project_File table.

The User table will contain the following attributes:

id (Integer) - a unique ID that represents each user

first_name (String) - the user's first name

last_name (String) - the user's last name

email (String) - a unique tufts email address

password_digest (String) - the encrypted user's password

updated_at (DateTime) - the time that the user's information was last updated in the database

created_at (DateTime) - the time that the user was added to the database

The Project table will contain the following attributes:

id (Integer) - a unique ID that represents each project

title (String) - the title of the project

department (String) - the Tufts representation of the department (i.e. COMP)

professor_last_name (String) - the last name of the professor teaching the course

keywords (String) - list of keywords relating to the project

user_id (Integer) - the ID of the user who created the project

updated_at (DateTime) - the time that the user's information was last updated in the database

created_at (DateTime) - the time that the user was added to the database

The Project_File table will contain the following attributes:

id (Integer) - a unique ID that represents each project file

project_id (Integer) - the ID of the project which the project_file belongs to

file (Attachment) - the project file

updated_at (DateTime) - the time that the user's information was last updated in the

database

created_at (DateTime) - the time that the user was added to the database

2. Models

The database tables will correspond to a User model, a Project model, and a Project_File model.

The User model will have the following validations:

first_name - must be present and have a length under 20 characters

last_name - must be present and have a length under 30 characters

password - must have a length longer than 6 characters

email - must be present, unique, case insensitive, and contain @tufts.edu

Note: Further password validation will occur using the has_secure_password method

The Project model will have the following validations:

title - must be present and have a length under 50 characters

department - will be selected from a dropdown list

professor_last_name - must have a length under 30 characters

keywords - must have a length under 200 characters

user_id - must be present

The Project_File model will have the following validations:

file - must be present, must be a pdf or mp4 file, and must be less than 20 MB

project_id - must be present

3. Database relations

Each user can have zero to many projects associated with that user. This allows each user to upload multiple projects.

Each project may only have one author. This author must be the user who uploaded the project.

Each project may have one to many project files associated with that project. This allows for the case where a project contains multiple components such as a video and text document.

IV. Searching Component

The searching component will allow the user to filter projects based on title, department, professor, and keywords. Once the search button is pressed, the full list of projects is filtered based on what fields are filled in by the user on the home page. The filtered projects are retrieved from the database by querying each attribute specified. If a field is left empty, then there will be no restraints on the projects retrieved for that

attribute.

The searching component will not be restricted to any users. This feature will be available to anyone accessing the site, regardless of whether or not they have an application account.

V. Static Pages

The following static pages will be used for the web application:

/home - This is the home page. It will allow users to search the project catalog by specifying fields specific to their search such as the title, professor, department, and keywords that correspond to the project(s) they are looking for.

/about - This page talks about the application and its capabilities

/contact - This page gives contact information in the case that a user encounters issues with the website.

VI. User Signup/Login

In order to upload a document to the website, it is required that the individual create an account on the website.

1. User Signup

User signup will require that the password be at least 6 characters in length. Password storage and authentication will require the use of the bcrypt gem and will use Ruby on Rails' has_secure_password method. This method stores a hashed password_digest field that represents the password. This guarantees that the user's password is not be exposed within the database and therefore cannot be leaked to any adversary.

2. User Login

Users will be able to log in to the site using Ruby on Rails' authenticate method. This method hashes the user's password and checks to verify that the entered account name and hashed password match with an account name and password_digest stored within the database. If the user has been authenticated, then the user will be able to access their account details and upload files to the site.

VII. Project Upload

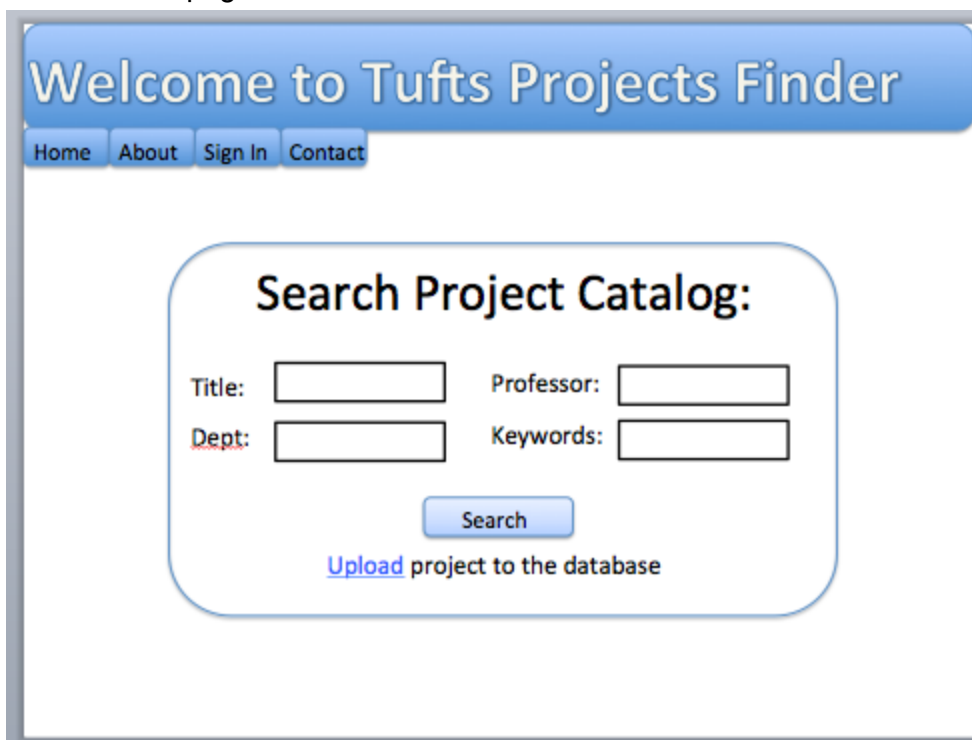
Once signed in to the website, user's will be able to upload their project files to the website so that others can search the project database. Project upload will use the Ruby

paperclip gem.

Individuals will upload their files when adding a project to the database. The project upload feature will use paperclip's `has_attached_file` method and will validate the attachment content type so that the file is either of type pdf or mp4. These file types were selected because they are safe file types that will not execute harmful data. They are also common file formats for presenting both text and audio/movie data. Almost all projects should be able to be represented by these file types. File size will also be restricted to less than 20 MB. This will prevent users from uploading too much data that slows down the web service or wastes server space.

VIII. Views

Home page view:



Welcome to Tufts Projects Finder

Home About Sign In Contact

Search Project Catalog:

Title: Professor:

Dept: Keywords:

[Upload](#) project to the database

Projects page view:

Tufts Project Finder

[Home](#) [About](#) [Sign In](#) [Contact](#)

Date	Title	Author

Sign in View:

Tufts Project Finder

[Home](#) [About](#) [Sign In](#) [Contact](#)

Sign In

Tufts Email:

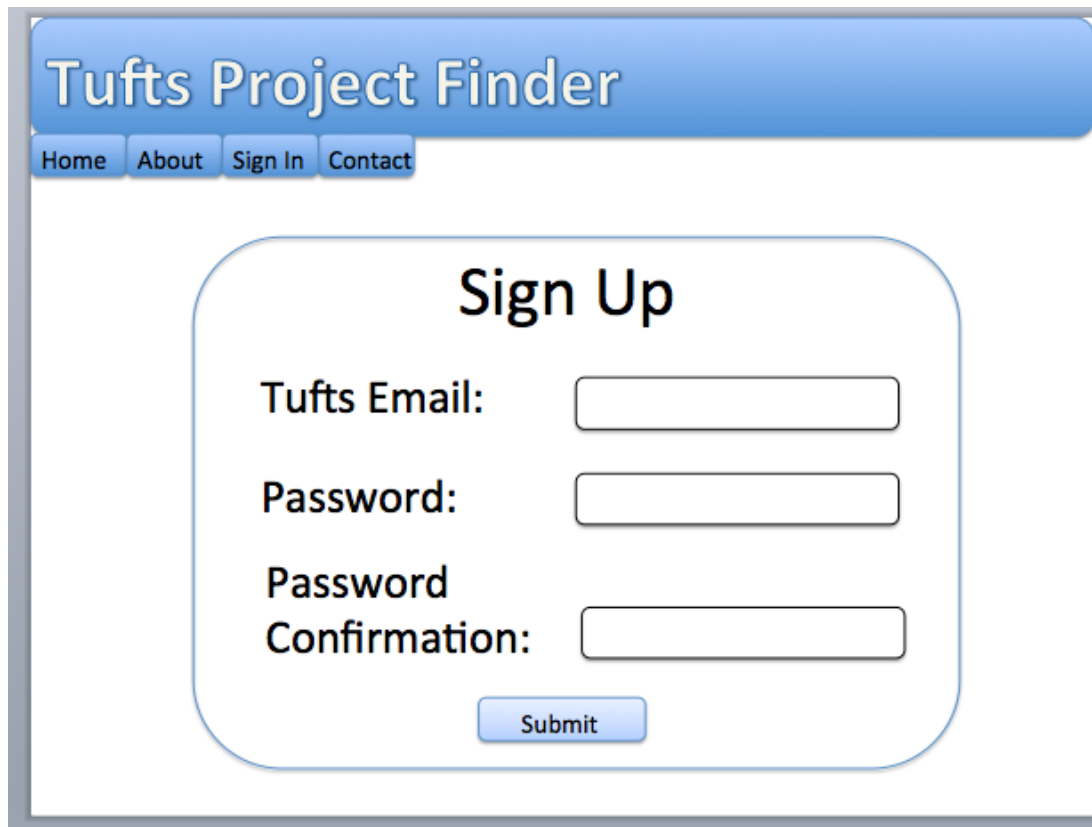
Password:

Note: You may only upload documents when signed in.

[Forgot Password?](#)

[New User?](#)

Sign up View:



The image shows a web application interface for 'Tufts Project Finder'. At the top, there is a blue header with the title 'Tufts Project Finder' in white. Below the header is a navigation menu with four items: 'Home', 'About', 'Sign In', and 'Contact'. The main content area features a 'Sign Up' form with a rounded rectangular border. The form contains three input fields: 'Tufts Email:', 'Password:', and 'Password Confirmation:'. Each field is followed by a white rectangular input box. Below the input fields is a blue 'Submit' button.

IX. Controllers

- Projects controller: The project controller will have show, new, and create methods. The show method will allow a user to see a project file when they click on a specific project from the projects page view. The new method will allow a user to create a new project. The create method will allow a user to upload a new project to the database.
- User Controller: The user controller will have show, new and create methods. The show method will be used when users information needs to be shown. The new method will allow a new user to be created. The create method allows a user to be uploaded to the database.
- Search Controller: The search controller will have a show method. This method will allow the filtered results to be seen on the home page.

X. Testing

This web application will be built using test driven development with Rspec and Capybara. This will help guarantee the correctness of the components of the web interface. Test

driven development will be used during all components of the website creation.

The intuitiveness of the user interface will first be tested by examining wire frames. Once these wire frames are constructed, they will be discussed and analyzed by Tina Woolston (the project advisor), other individuals working at the Tufts Office of Sustainability, and random members of the Tufts Community. After the first round of analysis has occurred, modifications to the wire frames will be made and construction of the actual user interface will begin. After completing construction of the user interface, a second round of analysis on the user interface will occur. This will be completed by the same individuals as mentioned before using new random members from the Tufts community. Modifications will be made according to this feedback.