CIS 435

Capstone Project

Nazariy Dumanskyy



Table of Contents

[Problem Statement 3](#_Toc469765990)

[Specifications 3](#_Toc469765991)

[User Responsibilities 3](#_Toc469765992)

[Analysis 3](#_Toc469765993)

[Design 3](#_Toc469765994)

[Use Cases 4](#_Toc469765995)

[UML Diagram 4](#_Toc469765996)

[Activity Diagrams 5](#_Toc469765997)

[UI Design Prototype 5](#_Toc469765998)

[Home Page 5](#_Toc469765999)

[Resume Page 6](#_Toc469766000)

[Post Page 6](#_Toc469766001)

[Conclusion 6](#_Toc469766002)

# Problem Statement

Design a community resume review website where users anonymously upload their resumes based on the career that they are in. Once uploaded a community of users would go in and anonymously comment and give feedback regarding what can be improved on the resume.

# Specifications

1. Home page where the user can view all the resumes that were submitted
   1. User is able to filter all the resumes by the career category
   2. Along with the resumes, user will see a few of the comments posted
   3. A link is provided to a resume page to view more comments and comment.
2. Resume page where a user can view an individual resume
   1. User can see all of the comments posted
   2. User can comment
3. Post page where a user can post his/her resume to the website
   1. A user can select a career category
   2. A user can browse for the pdf file of his resume

# User Responsibilities

1. Nazariy Dumanskyy
   1. Database Designer, Front-End Designer, Developer

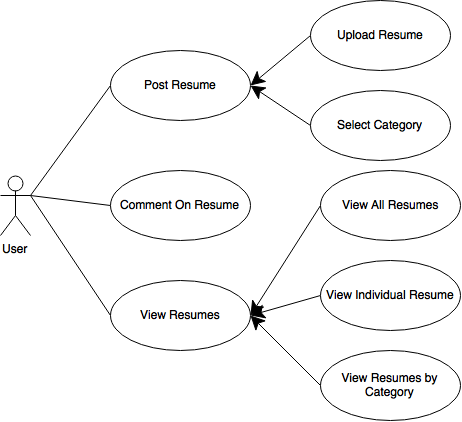
# Analysis

Picturing how the website is going to work, there are three main functionalities: upload resume, comment on a resume, and finally view resume. I first need to find out how to save and display PDF files. With that done, my goal is to start the coding process. Initially I need to divide all of my development into an MVC type framework structure, where the model contains functions to attain the information, view contains the page components, and controller binds that data to the view.

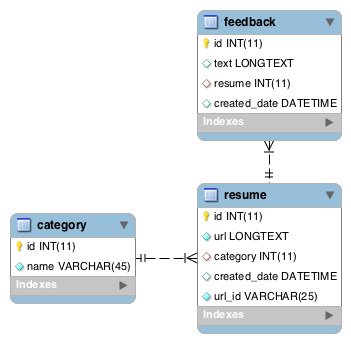
# Design

* To view the pdfs, I will be using pdf.js.
* Based on the data needed there really need to be three table entities
  + Resume entity to save the urls of the pdf files
  + Feedback entity to save all of the comments
  + Category entity to save all of the categories
* All of the resumes will be saved locally on the server with a unique name appended to it
* PDO will be used to query the database due to its simplicity and understanding
* Any form submission will be validate with javascript

# Use Cases



# UML Diagram

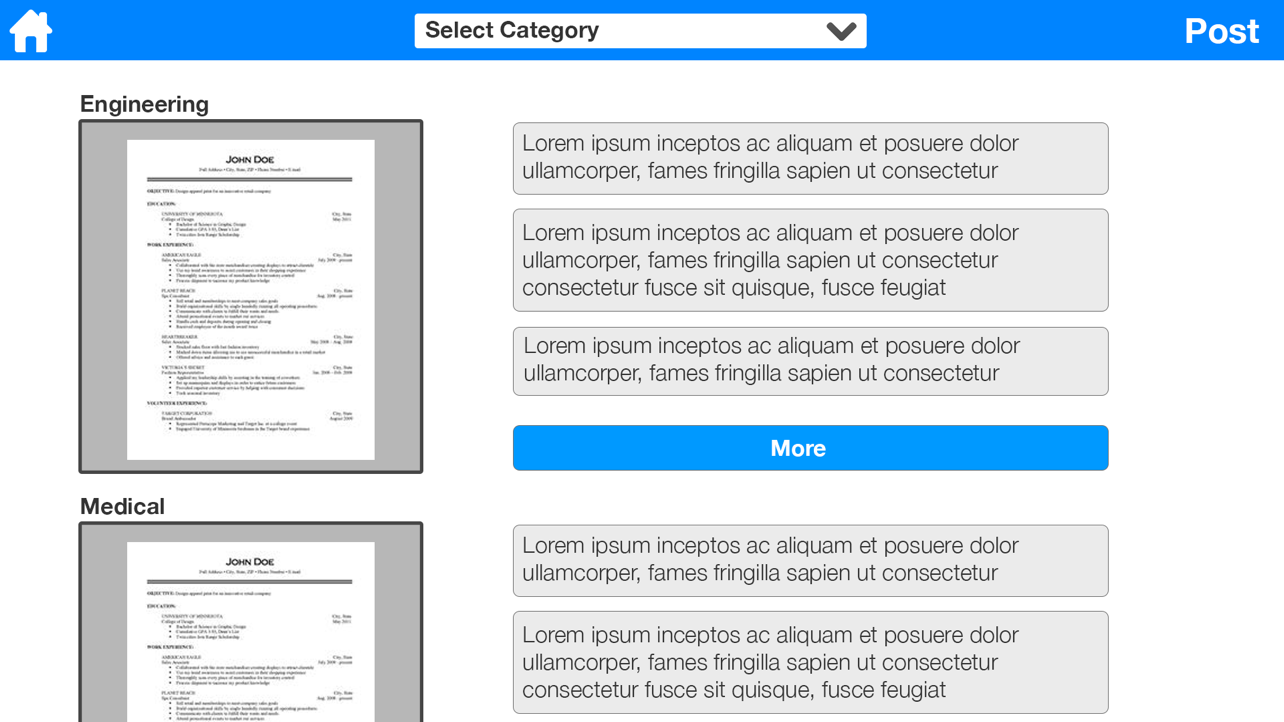


# Activity Diagrams

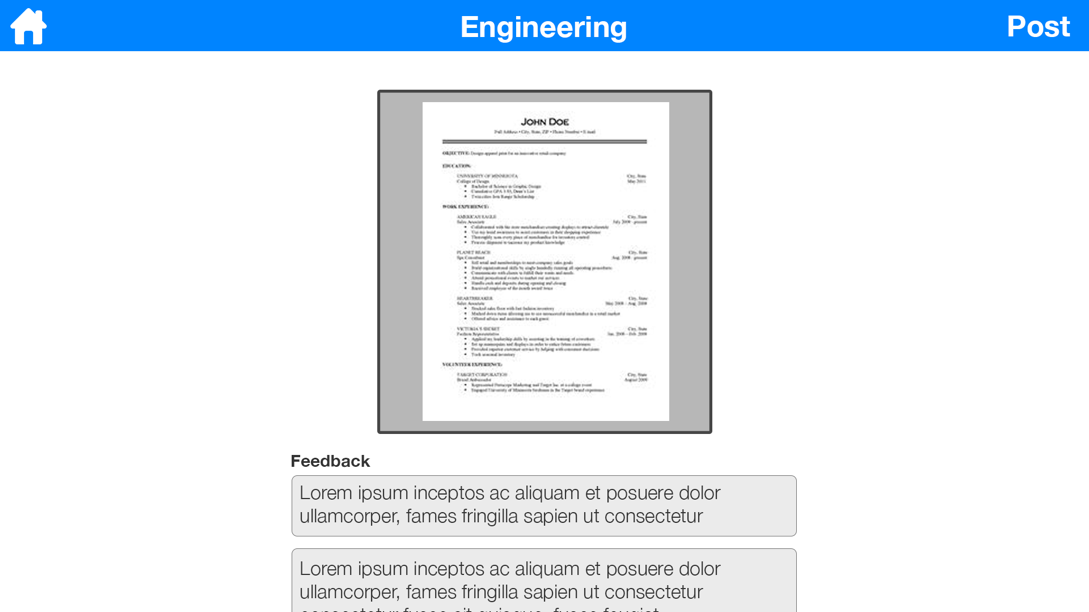
|  |  |  |
| --- | --- | --- |
| ../../../Downloads/Activity%20Diagram.png | ../../../Downloads/Activity%20Diagram-2.png | ../../../Downloads/Activity%20Diagram-3.png |

# UI Design Prototype

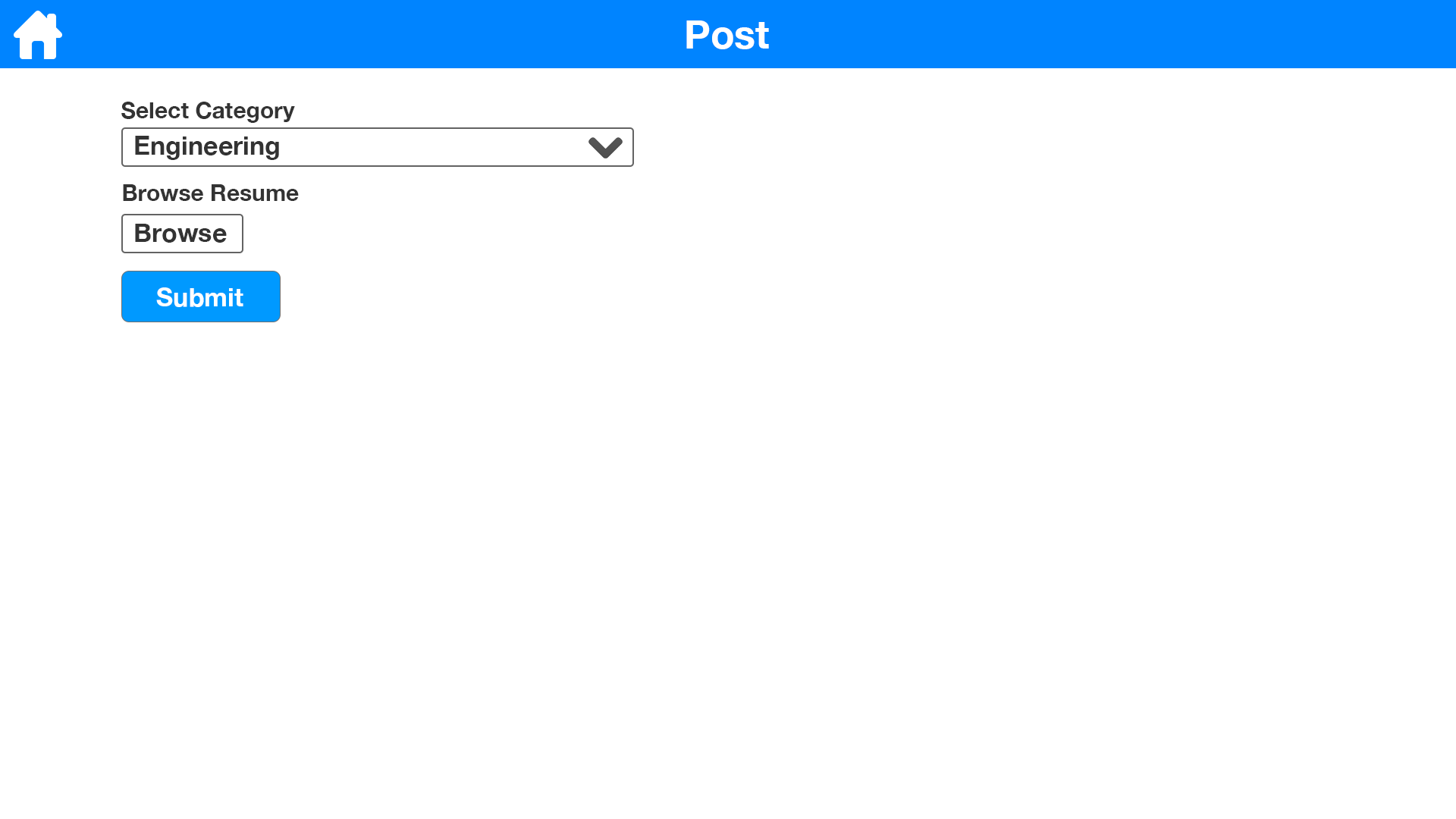
### Home Page



### Resume Page



### Post Page



# Conclusion

Overall, this was a very interesting project to do. I learned many things and solved many problems.

There were three challenges that faced me in this project:

1. I decided to use PDO for talking to the database and that was very interesting to learn.
2. I also tried to structure this project as an MVC project: model, view, controller. I have never done something like that in PHP, and I’m glad I tried.
3. I found out how to display pdfs using pdf.js. It really is a great framework, however it took some trial and error to get it running.

In addition to that, I learned how to prevent a form from being submitted twice on page refresh, and that was very interesting because I have never thought about a problem like this. Thus due to what have been learned here, I’m glad I got to do this project.