Software Requirement Specification

Problem Statement: A few sentences to describe the problem you are trying to solve, i.e., justify why this software is needed.

Finding the right course at the start of the semester can be a really challenging task for many. Furthermore, the task is more challenging for freshers who might have no contact with alumni, their peer or the teachers. JHU provides a brief description of the courses however that sometimes that's not enough and can be very confusing. How exactly can one base their entire future and the potential differences of offered courses, if all they're left with is that. How can one accurately assess the difficulties of a course if they are forced to message individuals, grades above them, to get the entire gist of a semester long class through one source instead of a pipeline. How is somebody supposed to know how to prepare if they are forced to take the two weeks that the school offers you to get the grasp of a class, lose said grasp of a class, and drop, just to repeat the cycle again but now with a different subject and no way out. There is a need for an application which can analyze the reviews, extract relevant information and provide with charts and graphs which can explain various factors like Assignment/Syllabus diffculty etc.

c.

Potential Clients: Who are influenced by this problem and would benefit from the proposed solution? (i.e. the potential users)

1. All students in JHU who would like to learn more about the courses they are interested in.
2. All students in JHU who would like to share their experience or opinion with others.
3. Professors who teach the course could add their own opinions on student evaluations.

Proposed Solution: Write a few sentences that describe how a software solution will solve the problem described above.

The proposed solution is to build a responsive web application which has a sentiment analysis engine which can be accessed using a mobile or a desktop computer. The plan is to initially give students the chance to splurge all the details they want to give about a class by touching on a variety of different sectors of what would compile to make up the complete value the course has to offer, and a summary of the course as well, once the data is compiled on the student's end it is shunted back to the server to perform some computations on the text and generate either a sentiment or a detaild analysis of the course through graphs and charts.

For students who want to learn more information about the course they are interested in, they can search with keywords and filters and even post a question which has not been answered yet. Based on the keywords, responses and overall gratifying or grating notions of the class the information will then be parsed through potentially a variety of methods, any student can access the reviews for any course. The professor of a particular course can review and share his thoughts about the course he would teach. Their review will have a higher priority than the student reviews. The reviews can be upvoted or downvoted depending on how helpful they are. The most helpful reviews will have a higher priority and will show up on top of the screen.

Functional Requirements: List the (functional) requirements that software needs to have in order to solve the problem stated above. List these in role-goal-benefit format. It is useful to try to group the requirements into those that are essential (must have), and those which are non-essential (but nice to have).

Must have

1. Only persons affiliated to JHU can login.
2. The search bar should display all the courses.
3. Reviews should be visible for everyone - even guests who do not login).
4. The reviews should be text-based as well as numerical.
5. Only previous or current students should be allowed to give a review.
6. Charts and Graph analysis

Nice to have

1. An in-depth analysis of the reviews.
2. A review voting system to put the most helpful reviews on top.
3. Professor's own reviews may be displayed.
4. Recommended Courses using a Recommendation Algorithm which displays the popular courses as well as the courses that suit the Student.
5. Keyword detection. Before a comment is successfully posted, the app checks for blocked keywords to avoid things like abuse.

Continues on the next page

Non-functional Requirements:

1. No huge load time
2. Preferences don't bog each other down
3. Cross-Compatibility with Different Browser systems
4. System meets the WCAG 2.1
5. Security of the users are paramount, reviews can be anonymous if need be.
6. System will perform without error in most cases

Software Architecture & Technology Stack: Will this be a Web/desktop/mobile (all, or some other kind of) application? Would it conform to specific software architecture? What programming languages, frameworks, databases, …, will be used to develop and deploy the software?

Website Application development; the most preferable is using the MERN stack
Language: Python for Data Analysis and Machine Learning.

Architecture: Client - Server Architecture.

Similar Apps: List a few similar applications to the one you are developing. Don't be eager to conclude no similar app exists! There is always something similar to what you are building! Finding those will help you to better specify your project. You must be prepared to explain how your app is different from the existing ones.

Rate My Professor: Flaws include the fact that it only tacks on information about the professors but professors don't always teach the exact same class and they're teaching styles while maybe staying the same can also differ with harder or easier subjects

While JHUReviews aims to do the same as our project goals. Anyone can review a course and also, it does not display the User's own profile. User can only search using various filters and give out reviews in the form of rating from 1 to 5. Any worded review is analysed by the website admin and summarised and averaged with the other reviews. There is no proper analysis regarding the various aspects of the course.