Software Requirements Specification

for

FACULTY EVALUATION THROUGH STUDENT FEEDBACK

Version 1.0 approved

Prepared by

Hemand K P
Abhishek V V
Akshay Raj
Devadath

Universal Engineering College

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

Introducing a project centered on advancing teaching evaluation methods through feedbacks, this initiative aims to revolutionize how feedback from students is categorized and utilized. By implementing an automated system capable of discerning feedback into positive, negative, or neutral categories, the project seeks to provide educators with actionable insights for refining their teaching approaches. Through the provision of clear and tailored feedback derived from student responses, this endeavor endeavors to empower teachers in enhancing their methodologies and ultimately fostering an environment conducive to enriching teaching and learning experiences.

1.1 Purpose

The project aims to enhance teaching evaluations by sorting feedback into positive, negative, or neutral categories. Its goal is to provide clear feedback from students to help teachers improve their teaching methods.

1.2 Document Conventions

This document follows the IEEE format standard (IEEE Std. 830-1998).

1.3 Intended Audience and Reading Suggestions

Faculty evaluation through student feedback is targeted towards educators, administrators, researchers, and policymakers within the realm of higher education. Readers interested in enhancing teaching effectiveness and evaluating faculty performance will find value in exploring this subject. Researchers seeking insights into innovative methods for assessing teaching effectiveness through student perspectives will also find this topic relevant.

1.4 Project Scope

Collaborating with schools, colleges, or universities to implement our project in their existing feedback systems, allowing for real-time analysis of student feedback. Involving not only faculty members and students but also other stakeholders such as academic advisors, department heads, and alumni in the feedback analysis process to provide a more holistic understanding of teaching effectiveness and student learning experiences.

1.5 References

1)"SENTIMENT ANALYSIS OF STUDENTS COMMENT USING LEXICON APPROACH". KHIN ZEZAWAR AUNG,NYEIN NYEIN MYO[University of Computer studies,Mandalay,Myanmar.] (2017)
https:scholar.google.comscholarhl=en&as_sdt=0%2C5&q=sentiment+analysis+of+students+comment+using+lexicon+approach+&btnG=

2)"Sentiment Analysis of Students Feedback Using Lexicon Based Method and Hybrid Machine Learning Method".
Shital A Patil, Krishnakant P Adhiya, Girishkumar K Patnaik
(2023)
https://ijisae.org/index.php/IJISAE/article/view/4498/3172

3)"Using Sentiment Analysis to Explore Student Feedback: A Lexical Approach".
(2023)
Rdouan Faizi ,Mohammed V University in Rabat, Rabat, Morocco
https://doi.org/10.3991/ijet.v18i09.38101

2. Overall Description

2.1 Product Perspective

The automated teaching evaluation system will serve as a comprehensive tool for educational institutions to streamline the process of collecting and analyzing student feedback. It will integrate seamlessly with existing feedback mechanisms, enhancing their capabilities by automating the categorization of feedback into positive, negative, or neutral categories

2.2 Product Features

- *Automatically categorize student feedback into positive, negative, and neutral categories.*
- *Provide educators with a user-friendly dashboard to access and visualize feedback.*
- *Ensure student feedback is secure and handled with care.*
- *Students would benefit from more effective teaching methods resulting from targeted improvements based on their feedback.*
- *Students can have confidence in the quality of education provided by knowing that their feedback is valued and acted upon.*
- *Automating the feedback analysis process saves time for faculty members, allowing them to focus more on teaching and student support.*

2.3 User Classes and Characteristics

The main characters in the system are faculty members and students. Faculty members, responsible for delivering instruction, require access to feedback insights and actionable recommendations for enhancing their teaching methods. Students, enrolled in educational programs, seek easy-to-use feedback mechanisms that assure their feedback is valued and acted upon. Transparent communication channels with faculty members are essential for students to provide meaningful input on their learning experiences.
1. Faculty Members:
- Characteristics: Educators responsible for delivering instruction.
- Needs: Access to feedback insights, actionable recommendations for improving teaching methods, and tools for tracking their performance over time.
2. Students:
- Characteristics: Individuals enrolled in educational programs providing feedback on their learning experiences.
- Needs: Easy-to-use feedback mechanisms, assurance that their feedback is valued and acted upon, and access to transparent communication channels with faculty members.

2.4 Operating Environment

WINDOWS
Windows is a graphical operating system developed by Microsoft. It allows users to view and store files, run the software, play games, watch videos, and provides a way to connect to the internet. It was released for both home computing and professional works. Microsoft introduced the first version as 1.0. It was released for both home computing and professional functions of Windows on 10 November 1983. Later, it was released on many versions of Windows as well as the current version, Windows 10.In 1993, the first business-oriented version of Windows was released, which is known as Windows NT 3.1. Then it introduced the next versions, Windows 3.5, 4/0, and Windows 2000. When the XP Windows was released by Microsoft in 2001, the company designed its various versions for a personal and business environment. It was designed based on standard x86 hardware, like Intel and AMD processor. Accordingly, it can run on different brands of hardware, such as HP, Dell, and Sony computers, including home-built PCs

2.5 Design and Implementation Constraints

The automated system may struggle to accurately interpret subjective feedback, as it relies on predefined categories and algorithms which may not capture the full nuance of student responses.

The effectiveness of the automated system may be limited by technical constraints such as processing power, bandwidth, or compatibility issues with existing systems.

Faculty members and students may be resistant to adopting the new system, particularly if they are accustomed to traditional evaluation methods or perceive the automated system as threatening their autonomy or privacy.

2.6 User Documentation

We create clear guidelines and documents to ensure the project's success,covering everything from system architecture to security policies,making sure we meet industry standards.

2.7 Assumptions and Dependencies

***Assumptions**:*
Users are comfortable with technology and have access to devices for system interaction. Faculty members are open to feedback and expect accurate categorization of feedback into meaningful insights.

Dependencies:
Successful implementation requires technical expertise for development, integration with existing systems, and compliance with regulatory standards. Collaboration from faculty and administrators is essential for effective implementation and sustainability.

3. System Features

3.1 System Features 1

3.1.1 Description and Priority

Provide a bench marking feature that compares faculty performance based on feedback against predefined criteria or against other faculty members within the same department or institution. This can help identify areas of improvement and promote continuous professional development.

3.1.2 Stimulus/Response Sequences

Stimuli for implementing performance bench marking in a faculty evaluation through student feedback system include defining clear evaluation criteria, selecting appropriate benchmarks or standards, generating comparison reports, enabling customization of bench marking parameters, providing real-time updates, integrating bench marking results into evaluation processes, ensuring accessibility, maintaining data security, offering user support, and ensuring scalability. These stimuli empower institutions to assess faculty performance effectively, identify areas for improvement, and foster a culture of continuous enhancement in teaching quality and student satisfaction

3.1.3 Functional Requirements

The functional requirements for performance bench marking in a faculty evaluation system include user authentication for authorized access, feedback data collection, normalization to ensure fair comparisons, benchmark selection options, generation of comparison reports, customization of criteria, real-time updates, integration with evaluation processes, accessibility considerations, data security measures, user support, and scalability to handle varying volumes of data. These features collectively enable the system to provide valuable insights into faculty performance, support decision-making processes, and promote continuous improvement in teaching effectiveness and student satisfaction

3.2 Usability and user Experience

3.2.1 Description and Priority

Usability and User Experience: Design the system with an intuitive and user-friendly interface to ensure ease of use for both students providing feedback and faculty members accessing evaluation results. Provide clear instructions and guidance throughout the feedback submission process, and incorporate features such as auto complete, error handling, and progress indicators to enhance the overall user experience. Additionally, ensure that the system is responsive and accessible across various devices and screen sizes to accommodate different user preferences and accessibility needs. A well-designed and user-friendly system enhances user engagement, encourages participation, and ultimately contributes to the success of the evaluation process

3.2.2 Stimulus/Response Sequences

Stimuli for enhancing usability and user experience in the faculty evaluation through student feedback system encompass intuitive interfaces with clear labels and visual cues, user-friendly feedback forms, responsive design for seamless access across devices, efficient submission processes, adherence to accessibility standards, instant feedback confirmation, robust error handling, progress tracking features, and personalization options. These elements collectively contribute to a positive user experience, encouraging active participation and ensuring accessibility for all users, ultimately facilitating the collection of valuable feedback data and promoting continuous improvement in teaching effectiveness and student satisfaction.

3.2.3 Functional Requirements

Functional requirements for a faculty evaluation through student feedback system include user authentication for secure access, course selection options for students, customizable questionnaire creation, feedback submission capabilities with anonymity options, analysis tools for data interpretation, feedback tracking for faculty, automated notifications, moderation features, performance benchmarking functionalities, data privacy measures, feedback response mechanisms, and seamless integration with other institutional systems. These requirements collectively ensure the system's effectiveness in collecting, analyzing, and utilizing feedback data to support faculty evaluation processes and enhance teaching quality

4. External Interface Requirements

4.1 User Interfaces

In this scenario, the dashboard offers two options: "STUDENTS" and "FACULTY." Students have the ability to write feedback, while faculty members can access the feedback submitted by students.

4.2 Hardware Interfaces

The main hardware interfaces for implementing the Digital out-pass system are given below:

- *Hard Disk – Minimum 5 GB.*
- *Processor speed- Minimum 1.4 GHz.*
- *RAM- Minimum 2 GB*

4.3 Software Interfaces

Operating system : Windows
Web Browsers : Google Chrome/Firefox/Internet Explorer
Front End: HTML, CSS, Java Script
Back End : Python 3.10 Django

Front End Tools

HTML: *HTML, known as Hypertext Markup Language, is the backbone of the web, providing a structured approach to building web pages. Through elements enclosed in angle brackets, HTML defines the content and its organization, encompassing a wide range of functionalities from headings and paragraphs to links and images. The document structure typically includes a declaration, root, head, and body sections, with attributes augmenting tags with additional information or behavior modifications. HTML5 introduced semantic elements like headers, navigation bars, main content areas, and footers, enhancing document clarity and accessibility. As the foundation of web development, HTML enables content dissemination and user interaction, facilitating dynamic functionality through integration with JavaScript and styling with CSS. Validation tools ensure compliance with W3C standards, verifying syntactic correctness and adherence to best practices, thereby ensuring robust and web content creation.*

CSS*: CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in HTML or XML. It controls the layout, colors, fonts, and other visual aspects*

of web pages, allowing for separation of content and presentation. CSS uses selectors to target specific elements and apply styles, and it operates on a cascading principle, where styles can be inherited or overridden based on specificity and order of application.

JavaScript:*JavaScript is a high-level, interpreted programming language primarily used for adding interactivity and dynamic behavior to web pages. It enables manipulation of HTML and CSS, handling user interactions, and asynchronous communication with servers (Ajax). JavaScript is supported by all modern web browsers and can be used both on the client-side (in the browser) and server-side (with platforms like Node.js). It features a rich ecosystem of libraries and frameworks such as React, Angular, and Vue.js, making it versatile for a wide range of web development tasks.*

Back End Tool

Python:*Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python's extensive standard library provides modules and functions for a wide range of tasks, from web development and data analysis to machine learning and artificial intelligence. It is widely used in various fields such as web development (with frameworks like Django and Flask), scientific computing, automation, and scripting. Python's clear syntax and large developer community make it an excellent choice for both beginners and experienced programmers.*

Django:*Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It follows the Model-View-Template (MVT) architectural pattern, which is similar to Model-View-Controller (MVC). Django's "batteries-included" philosophy provides built-in features for authentication, URL routing, database ORM (Object-Relational Mapping), template engine, and more, enabling developers to focus on building their applications rather than reinventing the wheel. It emphasizes DRY (Don't Repeat Yourself) and convention over configuration principles, making it suitable for building scalable, maintainable web applications.*

4.4 Communications Interfaces

Hypertext Transfer Protocol (HTTP) or its secure counterpart HTTPS are commonly used for communication between web browsers and web servers. We will use this interface for client-server communication in our web-based system. In our project, HTTP/HTTPS will serve as the primary communication interface between web browsers and the project's server. It enables the exchange of data, such as feedback submissions from students to the server, and the delivery of response data, such as feedback analysis results or dashboard updates, from the server to users' browsers.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- *Provide students with a user-friendly interface to submit feedback on their learning experiences, including options for categorizing feedback as positive, negative, or neutral.*

- *Implement an algorithm to automatically categorize feedback submissions into relevant categories, such as positive aspects of teaching, areas for improvement, or neutral comments.*
- *The website shall have the capability to store a large volume of feedback submissions.*

5.2 Safety Requirements

Implement strong user authentication mechanisms to verify the identity of users accessing the feedback system, preventing unauthorized access to sensitive data. This may include requiring users to use strong, unique passwords, for added security to prevent unauthorized access.

5.3 Security Requirements

Ensure that the feedback system's software components, including operating systems, web servers, and database management systems, are regularly updated with the latest security patches and fixes. This helps to mitigate the risk of security vulnerabilities being exploited by malicious actors and enhances the overall security posture of the system.

5.4 Software Quality Attributes

Performance: Specifies response times, throughput, and resource utilization.
Security: Outlines access controls, data encryption, and user authentication.
Usability: Describes user interface guidelines and ease of use.
Reliability: Specifies system up time, error handling, and fault tolerance.
Scalability: Addresses the ability to handle increasing amounts of data or users.
Maintainability: Describes code structure, documentation, and ease of maintenance.

6. Other Requirements

Appendix A: Glossary

Authentication:

The process of identifying an individual, usually based on username and password Cached A form of storing information/data, usually this data is repeatedly accessed.

CSS:

Cascading Style Sheets is a feature to give users and developers more control on how web site pages are displayed.

Database:

Is a structured collection of records or data that is stored in a computer system. In our system, this may pertain to flight records or user information.

Dynamic Links:

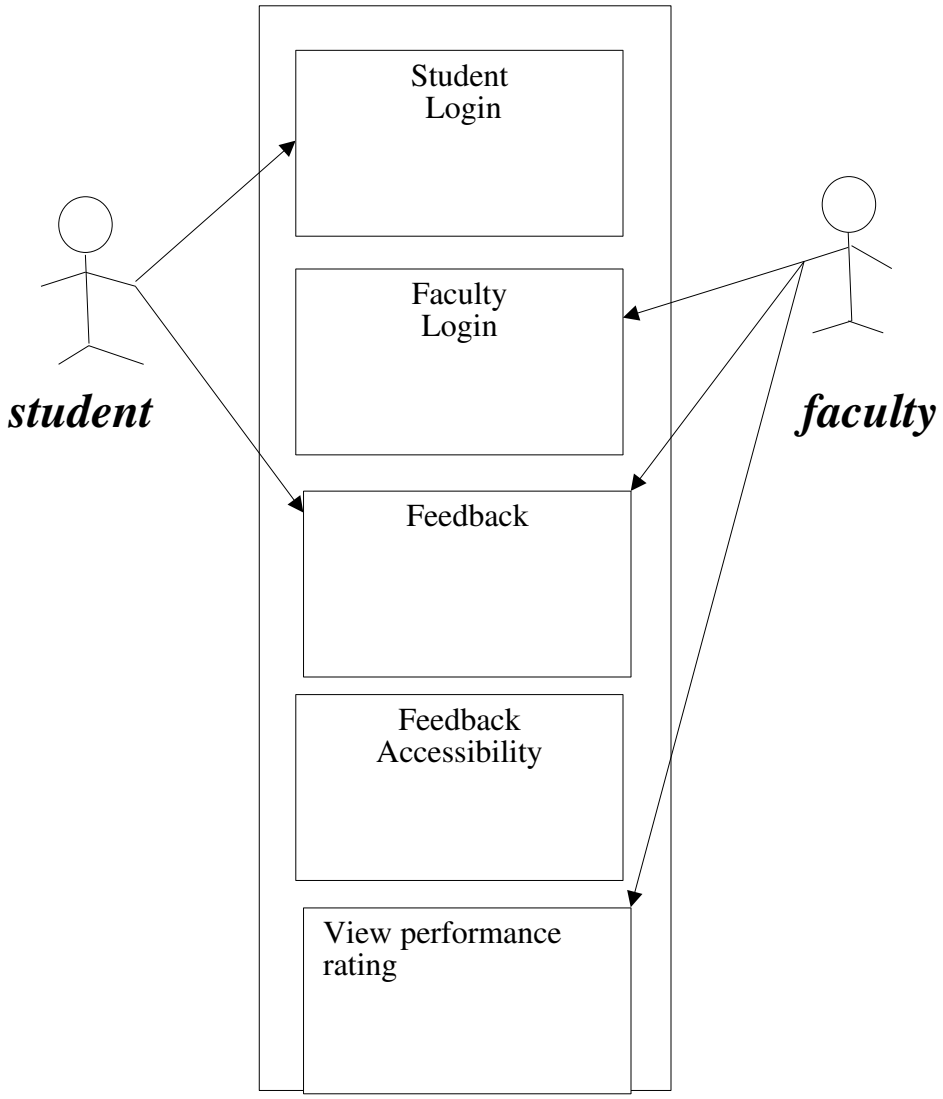
A pointer to a particular scope called during runtime

HTTP:

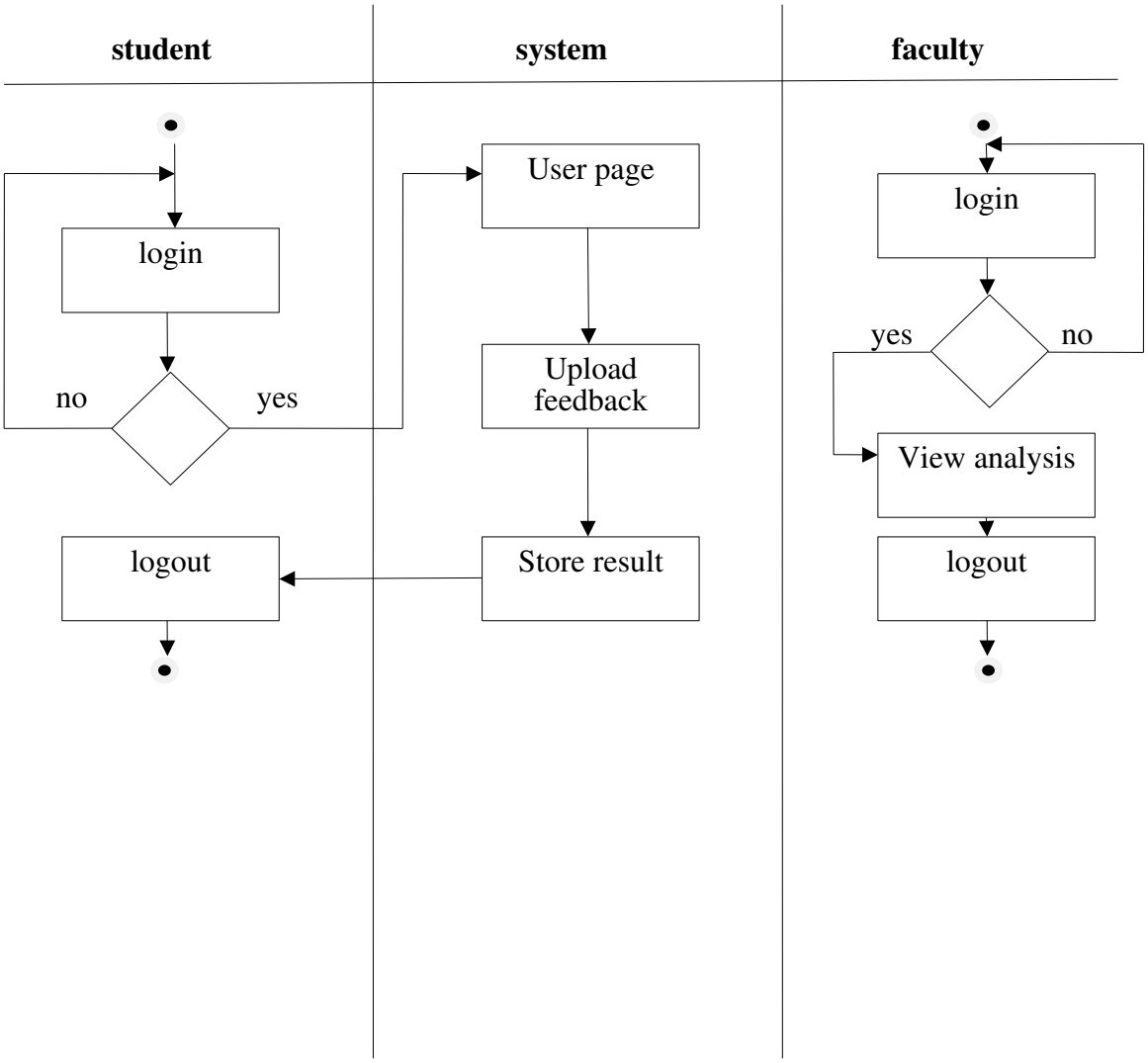
Hyper Text Transfer Protocol is the underlying protocol used by the World Wide Web. It defines how messages are formatted and transmitted and what actions should be taken in response to various commands.

Appendix B: Analysis Models

Use case diagram



Activity Diagram



Sequence Diagram