Shri Govindram Seksaria Institute of Technology and Science

Report on:

“Attendance and Class Assessment Marks Calculator”

Submitted to – Mr. Surendra Gupta , Ms. Ashwini Sharma

Department – Computer Engineering

Submitted by – Akshat Pandya

Enrollment Number – 0801CS221014 (Batch A3)

Email – akshatpandya27072004@gmail.com

Contact No. - +91-7389836702

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for the **Attendance and Class Assessment Marks Calculator**. This project is a Python-based application designed to simplify the process of calculating student attendance, lab attendance, assignment marks, quiz scores, and Class Assessment (CW) scores. The results are exported in Excel format for easy record-keeping and analysis.

1.2 Scope

The system is intended for use by educators to automate attendance and assessment score calculations based on customizable grading schemes. It will allow users to input data, apply a grading scheme, and generate final results in an Excel file. The application will be a desktop-based solution with a simple interface for user interaction.

1.3 Overview

This SRS document provides an overview of the system's functional and non-functional requirements, use cases, and system features, as well as technical specifications such as software and hardware requirements.

---

2. System Description

2.1 Product Functions

- Load and parse student data from an Excel file.
- Calculate attendance percentage for classes and labs.
- Calculate assignment and quiz scores using various grading schemes.
- Combine attendance, lab attendance, assignment, and quiz scores to calculate Class Assessment (CW) scores based on user-defined weights.
- Export final results with percentage symbols to an Excel file.
- Provide error handling for incorrect or missing data inputs.

2.2 User Characteristics

- **Primary Users:** Teachers, instructors, and educational administrators.
- **Technical Expertise:** Users are expected to have basic knowledge of file systems (loading and saving Excel files) but no programming experience.

---

3. Functional Requirements

3.1 User Interface (UI)

- The user interface should provide a simple way for users to:
  - Load student data from an Excel file.
  - Enter total classes and lab sessions held.
  - Define the grading scheme for assignments and quizzes.
  - Set weight distribution for each component (attendance, labs, assignments, quizzes).
  - Export final results to Excel format.
- The UI should be developed using the Tkinter library for simplicity.

3.2 Attendance Calculation

- The system should calculate attendance as a percentage:
  - **Formula:** (Classes Attended / Total Classes Held) * 100

3.3 Lab Attendance Calculation

- The system should calculate lab attendance as a percentage:
  - **Formula:** (Lab Sessions Attended / Total Lab Sessions) * 100

3.4 Assignment & Quiz Score Calculation

- Users must be able to choose from the following grading schemes for assignments and quizzes:
  - **Best Of All**: Highest score taken for final calculation.
  - **Average**: Average of all scores.
  - **Relative Grading**: Scores relative to the best performance among all students.

3.5 Class Assessment Calculation

- The system should calculate Class Assessment (CW) scores by combining attendance, lab attendance, assignment, and quiz scores using user-defined weights.
- Example weight distribution: 20% attendance, 10% lab attendance, 35% assignments, 35% quizzes.
- CW scores are calculated using user-defined weights
- Formula:

$$\text{CW}=(\text{Attendance}\times\text{WeightAttendance})+(\text{Lab Attendance}\times\text{WeightLab})+(\text{Assignment}\times\text{WeightAssignment})+(\text{Quiz}\times\text{WeightQuiz})$$

Example: If the weight distribution is 20% attendance, 10% lab attendance, 35% assignments, and 35% quizzes:

$$\text{CW}=(20\%\times\text{Attendance})+(10\%\times\text{Lab Attendance})+(35\%\times\text{Assignment})+(35\%\times\text{Quiz})$$

3.6 Excel Export

- The final results should be exported in an Excel format with appropriate formatting (including percentage symbols for scores).
- The export functionality should handle large datasets (up to 100 students) efficiently.

3.7 Error Handling

- If any data is missing or incorrectly formatted, the system should display an appropriate error message and guide users to correct the input.

4. Non-Functional Requirements

4.1 Performance

- The system should be able to calculate and export results for up to 100 students in under 5 seconds.

4.2 Usability

- The system must have an intuitive user interface that is easy to navigate for non-technical users.
- The interface should be simple enough for teachers and administrators with no programming knowledge.

4.3 Security

- Since the system operates locally, basic security measures like file validation and error handling should be implemented to protect the integrity of student data.

4.4 Scalability

- The system should be capable of handling large datasets with up to 100 students without significant performance degradation.

5. System Features

5.1 Input File Handling

- The system should accept Excel files with student data.
- The file must include fields for student names, attendance records, assignment scores, and quiz scores.

5.2 Grading Scheme Flexibility

- The system should allow users to choose and apply different grading schemes to suit their specific requirements.
- The grading scheme selected should be applied consistently across all students.

5.3 Export Functionality

- After calculations are completed, the results should be exported in a well-formatted Excel file, with percentage symbols correctly applied where applicable.

---

6. System Design

6.1 Architecture

- The system will follow a modular architecture, where different components (attendance, assignments, quizzes) are handled independently and combined at the end to produce the final result.

6.2 User Interface Design

- The user interface will be implemented using **Tkinter**, a simple and lightweight Python GUI library. The main components of the UI will include:
  - File selection dialogs for loading input files and saving output files.
  - Text fields for user inputs such as total classes, lab sessions, and grading scheme selection.
  - Buttons for calculating and exporting results.

6.3 Data Flow Diagram

- **Step 1:** User loads an Excel file with student data.
- **Step 2:** User enters total classes, labs, and maximum marks for assignments/quizzes.
- **Step 3:** System processes the data, calculates scores, and combines them according to weights.
- **Step 4:** System exports the calculated results to an Excel file.

---

7. Software and Hardware Requirements

7.1 Software Requirements

- **Programming Language**: Python 3.x
- **Libraries**:
  - Pandas: For handling Excel file operations.
  - Tkinter: For creating a graphical user interface.
  - Openpyxl: For reading/writing Excel files.
- **Operating System**:
  - Windows 7 or higher, macOS, or Linux.

7.2 Hardware Requirements

- **Minimum Hardware**:
  - Processor: Intel i3 or equivalent.
  - RAM: 4 GB or higher.
  - Storage: 500 MB available space.

---

8. Testing

8.1 Input Format

- The input file must be in Excel format (.xlsx) with the following fields:
  - **Student Name**: Name of the student.
  - **Classes Attended**: Number of classes attended by the student.
  - **Total Classes Held**: Total number of classes held.
  - **Lab Sessions Attended**: Number of lab sessions attended.
  - **Total Lab Sessions**: Total number of lab sessions held.
  - **Assignment Scores**: Scores from various assignments (multiple columns).
  - **Quiz Scores**: Scores from various quizzes (multiple columns).

8.2 Output Format

- The output will be exported as an Excel file (.xlsx) with:
  - **Student Name**
  - **Attendance Percentage**

- **Lab Attendance Percentage**
- **Assignment Score** (based on the chosen grading scheme)
- **Quiz Score** (based on the chosen grading scheme)
- **Class Assessment (CW) Score**

8.3 Processing

- The system performs the following steps:
    1. Load input data from the Excel file.
    2. Calculate attendance and lab attendance percentages.
    3. Calculate assignment and quiz scores based on the chosen grading scheme.
    4. Calculate CW score using the defined weights.
    5. Export the results into an Excel file.

---

9. Conclusion

The **Attendance and Class Assessment Marks Calculator** is a robust tool designed to automate the calculation of student performance data, making the process faster, more accurate, and easier to manage. It provides a user-friendly interface and flexible grading schemes, helping educational institutions streamline their grading workflows while minimizing manual effort and errors.