Software Requirements Specification

for

GRACE MARK ALLOCATOR

Version 1.0 approved

Prepared by Pavan, Awez, Lokesh, Upendra and Pruthve

Aug 20, 2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of developing this application is to reduce the time and effort taken for calculating the grace marks to be allotted to a student after verification of the documents and proofs and calculate the grades accordingly. This reduces the need for cumbersome communication, manual verification, etc. and provides centralized information access. Faculty can also choose to accept/reject the documents submitted by students, giving them more autonomy and restricting plagiarism.

1.2 Intended Audience and Reading Suggestions

This product is a prototype for the Grace Mark Allocator and it is restricted within the college premises. This has been implemented under the guidance of a college professor. This project is useful for the students as well as to the faculties of a university.

1.3 Project Scope

Every year the tracking of paper publications, extra-curricular events and club events that provide students with grace marks and finding the right subject to which the marks can be added in order to increase the grade of the student in the best way possible is a very manually hectic job. We have designed a software that eases the tedious process of handling the allocation of grace marks to students belonging to a semester, impartially. To do so, we set rules that govern the generation of grace marks and constrain them. Grace marks are generated on the basis of the proofs that have been submitted and verified. The generated marks are then added to the existing marks and published.

2. Overall Description

2.1 Product Perspective

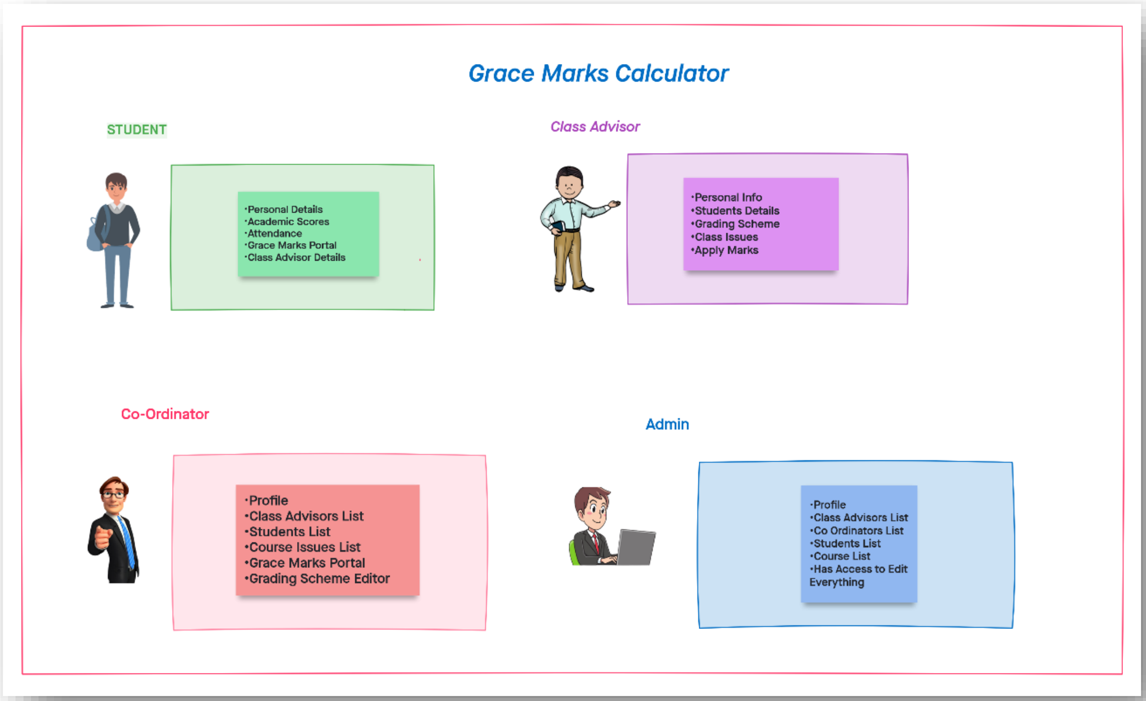
The online work environment has increased the need for collaboration like never before. Also manual verification and calculation is a cumbersome process. A Smart Grace Mark Allocator will facilitate easy and straightforward communication between student, faculty and exam department, removing any ambiguities. Students can track their document approvals; faculty can verify, accept/reject and allocate marks; and the exam office can assign the mark scheme and from the mark allocator find the best possible grades and publish them.

2.2 Product Features

This product will be scalable, flexible and will have a UI which is easy to be used by any type of users. The UI will be minimalistic. The website will be user friendly.

2.3 User Classes and Characteristics

The different users and their abilities can be understood by the below diagram:



2.4Operation Environment

This project will be a live website hosted on the Heroku platform.

- Operating System: Any
- Browser: Google Chrome, Mozilla Firefox, Microsoft edge
- Database: My Sql (Server-side)
- UI: HTML, CSS, Js (Client-side)
- Backend: Node Js (Server-side)

2.5 Design and Implementation Constraints

There are a few constraints, namely:

- We need to design a UI which has minimum load time, so the users with low internet speed are not affected.
- We need to make sure the UI design is simple, paving way for all users to operate with no training.

2.6 User Documentation

We will make a video tutorial on how to use the product and make sure of its availability on YouTube as well as our home page.

2.7 Assumptions and Dependencies

- The user will have basic English knowledge to access the website.

3. System Features

3.1 Description and priority

The server maintains details in the form of DB consisting of the name, e-mail, roll number, group the student belongs to, the mentor and their password in case of the user being student. Name, Department, Groups they are assigned as mentors, evaluators, and if they are the academic coordinators of that batch and the like, in case of a faculty. Of course, this project has a high priority because it is the need of the hour for all type of users (students, faculties).

4. Client server model

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server (also known as the back-end).

A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

5. External Interface Requirements

5.1 User Interfaces

The front-end is made up of HTML, CSS, Js.

5.2 Hardware Interfaces

- A computer with internet connection.

5.3 Software Interfaces

- The front-end is made up of HTML, CSS, Js.
- The back-end is made up of Nodejs and Ejs pages..
- The database is MySql.

5.4 Communications Interfaces

This project supports all types of web browsers.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

- The website must load in minimal time.
- The product must be scalable and flexible.

6.2 Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed-up log, up to the time of failure.

6.3 Security Requirements

There must be a middleware that does cleaning to the input data (data, username, password, files uploaded) so that there is no unwanted code executed on server to prevent the server from attacks like XSS, CSRF, NoSQL attack.

6.4 Software Quality Attributes

- **Availability:** The website must be running up at all times.
- **Integrity:** The data of a project (files, team members) must be safe and not be tampered by unauthorized people.
- **Maintainability:** The Admin must be able to transfer people from one team to another incase the department asks them to do so.
- **Reliability:** The evaluating faculty must be able to retrieve the documents of a team whenever they require it without it being tampered.

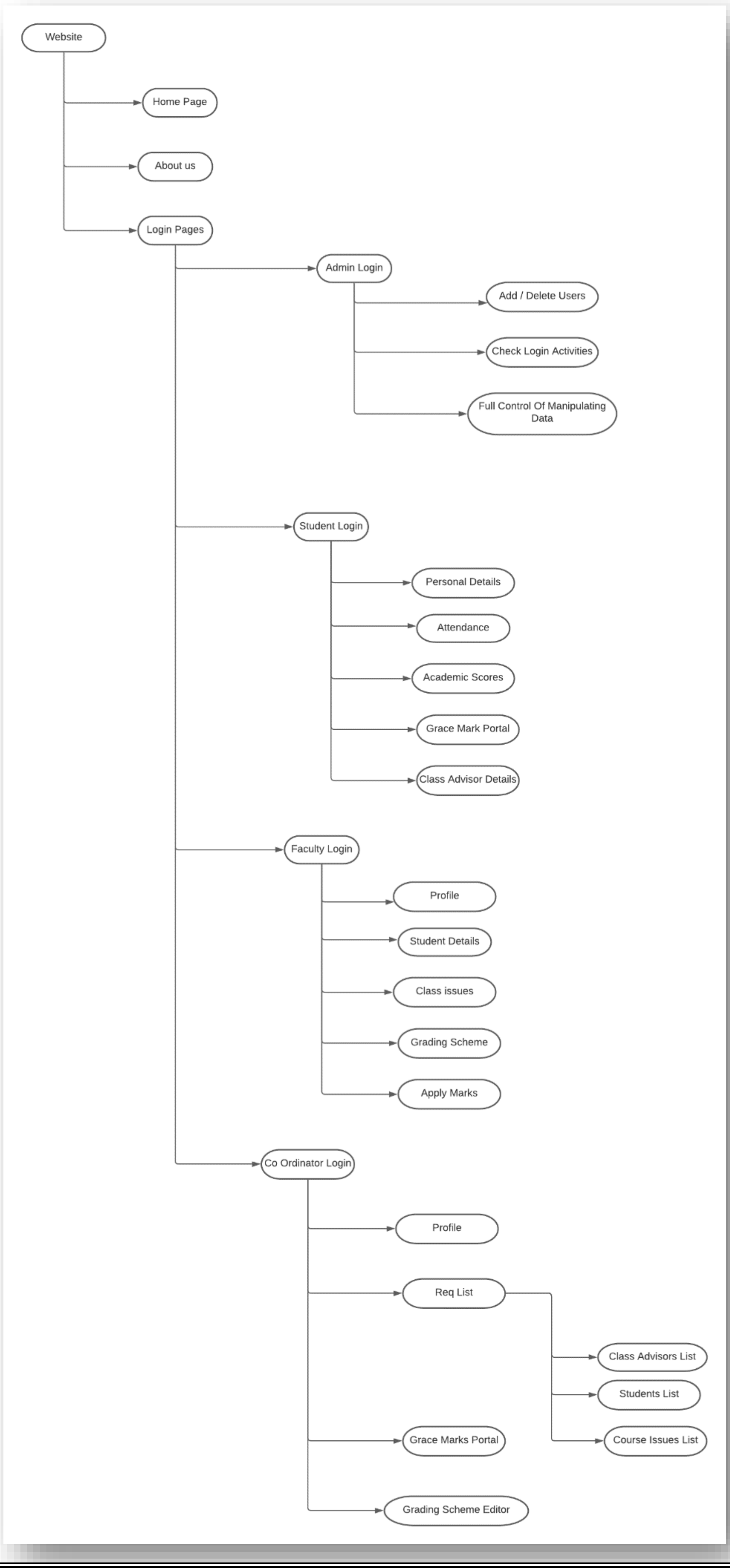
7. Other Requirements

Appendix A: Glossary

DB	Database
UI	User Interface
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
Js	Java Script
EJS	Embedded Java Script
Node Js	Node Java Script
SQL	Structured Query Language
MVC	Model, View, Controller

Appendix B:

Analysis Models


Website
Home Page
About us
Login Pages
Admin Login
Add / Delete Users
Check Login Activities
Full Control Of Manipulating Data
Student Login
Personal Details
Attendance
Academic Scores
Grace Mark Portal
Class Advisor Details
Faculty Login
Profile
Student Details
Class issues
Grading Scheme
Apply Marks
Co Ordinator Login
Profile
Req List
Class Advisors List
Students List
Course Issues List
Grace Marks Portal
Grading Scheme Editor