**Software Requirements Specification**

**for**

Student Management System

**Prepared by Rameen Rafiq, Laiba Fatima, Bharti, Hafsa Salman**

**FAST National University**

**04 May, 2023**

**Table of Contents**

**Table of Contents ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 3

2.6 User Documentation 3

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 4

**4. System Features 4**

4.1 Authentication 4

4.2 View Registered Courses 5

[4](#_4k668n3).3 View Attendance 5

4.4 View Marks 6

4.5 Calculate GPA 7

4.6 Calculate Fee Structure 8

**5. Other Nonfunctional Requirements 9**

5.1 Performance Requirements 9

5.2 Safety Requirements 9

5.3 Security Requirements 9

5.4 Software Quality Attributes 9

5.5 Business Rules 9

**Appendix A: Glossary 10**

**Appendix B: Analysis Models 10**

# Introduction

## Purpose

The aim of this project is to create a platform where the students can keep track of their academic activities such as attendance, marks, GPA, fee structure, and registered courses. Each student will first have to authenticate themself to proceed and view any of the options presented in the menu. The program contains all academic information for a student for their current semester at one place. This program will reduce the effort and time to maintain records for students manually and will quickly display the results of the selected option, which is not possible when done manually.

## Document Conventions

* Main Heading

Font: Arial

Size: 18

Style: Bold

* Subheading

Font: Arial

Size: 14

Style: Bold

* Body

Font: Arial

Size: 11

Style: none

## Intended Audience and Reading Suggestions

The audience intended to use this software will be administration staff, students, and teachers, and the development team.

## Product Scope

The project’s scope is confined to the basic features of any Student Management System such as viewing marks, attendance, fee, courses, and calculation of GPA. When any student’s GPA is calculated, a warning message will be displayed if the individual GPA is less than 1.0, and cumulative GPA is less than 2.0. It can also be used for a different set of students and courses. Any student can choose to login or logout from the system at any given time.

**1.5 References**

* <https://www.geeksforgeeks.org/>
* <https://www.javatpoint.com/>
* https://softwareengineering.stackexchange.com/

# Overall Description

## Product Perspective

The Student Management System has been designed to manage academic details and data for each student altogether at one place. Once the student ID and password has been authenticated, the menu will present several options to choose from. With each option, details of courses such as credit hours can also be viewed. The student also has the option to calculate GPA for only one course, or cumulative for all courses.

The product is independent, and not a part to be integrated in any larger system.

## Product Functions

The Student Management System is expected to provide the following functionalities:

* Login and Logout
* Authentication of students
* View registered courses
* Viewing attendance of different registered courses
* Viewing marks of different registered courses
* Calculate individual GPA
* Calculate cumulative GPA
* Calculate fee structure

## User Classes and Characteristics

The students must have knowledge of computers, should be proficient in English, and should be able to correctly enter the options asked by the program, such as ID, password and the option for the desired feature.

The program consists of one type of user, that is the student and can perform functions such as:

* authenticate themselves through provided login credentials
* view courses, attendances, marks, fee
* calculate individual and cumulative GPA

## Operating Environment

Operating System: Windows

Language: Java

Software: JDK/JRE and IntelliJ IDE

## Design and Implementation Constraints

The coding of this project has been done in Java, hence it is necessary to have JDK or JRE installed on the system and is supposed to work on a Windows operating system.

## User Documentation

Along with the product, the System Requirements Specification and the PowerPoint presentation will be delivered for ease in understanding the system.

## Assumptions and Dependencies

The number of students currently set is 4, additionally the program can only work when either one of the four students’ credentials are used. The software is also only limited to be used on systems containing the program.

It has been assumed that the login IDs and passwords have been created by administration members and communicated to the students to avoid security breaches. Moreover, since all student data has already been entered into the system, it can be assumed that this data has been fed at the end of semester, and no new marks will be entered or changes will be made.

# External Interface Requirements

## User Interfaces

The system provides an easy-to-use interface for users to enter their login ID and password, and to choose among options.

When a user enters the wrong ID, the system will cross-check and an output message will be displayed of the incorrect entered ID, similar will be the case for incorrect passwords. The following message highlighted in the color red will be displayed when incorrect ID/password is entered: “Invalid ID/password entered. Please try again!”

When an incorrect option from the menu is entered, another error message will be displayed, and the user will be prompted to re-enter a valid option. The following message highlighted in the color red will be displayed: “Please choose a valid option!”

The program also makes use of exception handling throughout the system where inputs have to be entered when any wrong argument is input by the user.

## Hardware Interfaces

Various interfaces for the product can be:

* Monitor: to display the program
* Mouse: to point to where the user wishes to
* Keyboard: to enter user inputs

**3.3 Software Interfaces**

The following interfaces are required to run the program:

* any Windows OS
* JDK/JRE, and a suitable IDE for Java such as IntelliJ

The system will first ask the user if they wish to login or to exit the program. When the user enters option 1, the system will proceed further and authenticate the user, on the other hand, if the user opts for option 2, it will exit the system.

The system will then input the student’s ID, password, and when the correct credentials are entered, the user will be required to enter the option of the feature they wish to use.

In the instance of choosing option 1, that corresponds to viewing registered courses, all the courses will be displayed.

When option 2 is chosen, which is to view course attendance, the user will be asked to enter the course code, and then the attendance of that particular course will be displayed.

For option 3, course marks for any chosen course will be displayed.

In option 4, the user will be asked to choose among calculating any individual course GPA, or cumulative GPA. If the user wishes to calculate individual course GPA, they will be prompted to enter the course code, and the GPA will be displayed. For cumulative, the overall GPA will be calculated and displayed.

When option 5 is chosen, the fee structure for that student will be calculated and shown.

## 3.4 Communications Interfaces

Connections to the system will be over TCP/IP connection.

The interface allows easy navigation between different program features and options.

The system is able to handle multiple user requests simultaneously without causing delay.

# System Features

The following are the features of the Student Management System:

## Authentication

**4.1.1 Description and Priority**

The authentication is the first and foremost feature of the Student Management System, with the highest priority, without which the program will not move further.

This feature will ask for student ID and password, and check against the credentials already stored in the system for verification purposes.

**4.1.2 Stimulus/Response Sequences**

* The program shows the WELCOME screen
* The user is asked to choose among logging in the system, or exiting
* The user selects the login option
* The user is asked to enter their student ID
* The ID is checked, if correct the system proceeds to ask for the password, else a message is displayed saying “ERROR! Invalid ID entered. Please try again”. In case for wrong ID, the user is prompted to enter their ID again
* The user is asked to enter password
* The password is checked, in case when the wrong password is entered a message is displayed saying “Incorrect password entered, please try again!”
* Upon successful authentication, further options are displayed for the user to choose from

**4.1.3 Functional Requirements**

REQ-1.1: User shall be able to login through their credentials

REQ-1.2: The system should authenticate login credentials against the data saved

REQ-1.3: The user is presented with a number of options for the authenticated student ID to choose from

An anticipated error condition could be entering alphanumeric or special characters when the user is expected to enter their student ID and password. In both the cases, the system should print an error message, and ask the user to enter their credentials for verification again. Additionally, the user can enter any other option other than the ones defined in the menu, the system should display an error message and ask the user to enter the option they are interested in again.

## View Registered Courses

**4.2.1 Description and Priority**

For any given student, this feature provides the list of all the registered courses for the current semester.

The student ID and name are displayed firstly, followed by columns of course code and names.

**4.2.2 Stimulus/Response Sequences**

* The user chooses option 1, which is to display all the registered courses for any student
* The system displays the student ID, and name for which the registered courses are being printed for
* A list of all the registered courses are printed. The course codes are left aligned and the corresponding course names are printed towards the right side.
* The user is asked to choose among the following options to proceed further: to continue with the same ID, to logout the current ID, or to exit the program

**4.2.3 Functional Requirements**

REQ-2.1: For any student ID, the registered courses code and names are displayed

This feature does not require any input from the user.

**4.3 View Attendance**

**4.3.1 Description and Priority**

When chosen, the attendance of any registered course will be displayed. The function will firstly display the list of registered courses, and ask the user to enter the course code of the subject attendance has to be viewed for.

After entering the valid course code, a statement will be displayed informing the course code, and name the attendance is being shown for. The total number of classes, classes attended by the student, and attendance percentage will also be displayed.

**4.3.2 Stimulus/Response Sequences**

* The user chooses option 2, which is to display attendance
* The list of registered course codes and names are displayed
* The user is asked to enter the course code of the subject they are interested to view attendance for
* The course code is validated, in the situation where incorrect course code is entered an error message is displayed saying: “Wrong input entered. Please enter a valid option”
* The function proceeds by printing: “Displaying attendance for [course code] [course name]”
* The total number of classes are displayed
* The number of classes attended are displayed
* The attendance percentage is calculated, and shown
* The user is asked to choose among the following options to proceed further: to continue with the same ID, to logout the current ID, or to exit the program

**4.3.3 Functional Requirements**

REQ-3.1: For any student ID, the attendance for their registered courses will be displayed

REQ-3.2: The user will be allowed to choose from any course to view attendance for

REQ-3.3: Total classes will be displayed

REQ-3.4: The number of classes attended by the student are shown

REQ-3.5: Attendance percentage of classes percentage are calculated and displayed

The user can be expected to enter the incorrect course code for which they want to view attendance for. The program should respond to the incorrect input by displaying an error message and ask the user to enter the course code again.

**4.4 View Marks**

**4.4.1 Description and Priority**

For any chosen course, this feature will display marks obtained by the student in any particular course. Firstly, the list of registered courses will be displayed, and the user will be prompted to enter the course code of the subject marks to be viewed for.

After entering the valid course code, a statement will be displayed informing the course code, and name the marks are being shown for. This will be continued by total marks, and the marks obtained in that student for any student.

**4.4.2 Stimulus/Response Sequences**

* The user chooses option 3, which is to display marks for the student for any chosen course
* The user is asked to choose among the registered courses they want to view marks for
* The list of course codes and course names are displayed
* The user enters the course code of the subject they are interested to view marks for
* The course code is validated, in the situation where incorrect course code is entered an error message is displayed saying: “Wrong input entered. Please enter a valid option”
* The course code and course name are printed for which marks will be displayed
* The total marks are printed
* Marks obtained by the student in that particular course are printed
* The user is asked to choose among the following options to proceed further: to continue with the same ID, to logout the current ID, or to exit the program

**4.4.3 Functional Requirements**

REQ-4.1: For any student ID, the marks for their registered courses will be displayed

REQ-4.2: The user will be allowed to choose from any course to view marks for

REQ-4.3: Total number of marks will be displayed

REQ-4.4: Marks obtained by the student in their chosen course will be shown

The user can be expected to enter the incorrect course code for which they want to view marks for. The program should display an error message and ask the user to enter the course code again.

**4.5 Calculate GPA**

**4.5.1 Description and Priority**

This feature has been divided into two further features: to calculate individual course’s GPA, or GPA of all courses combined.

When the option of calculating individual GPA is chosen, the list of course code and course names will be displayed for which GPA has to be calculated. The user will be asked to select among the courses. When the valid course code will be entered, the chosen course code, name, GPA and grade for that course will be displayed.

On the other hand, when the cumulative GPA option will be selected, the course name and course codes of all the registered courses will be displayed, followed by the cumulative GPA for the student.

**4.5.2 Stimulus/Response Sequences**

* The user chooses option 4, which is to calculate GPA
* The user is presented two options to choose from, 1 to calculate GPA for any course individually, and 2 to calculate combined GPA of all courses for any student ID
* The user enters the option they are interested in, in the instance when an incorrect option is entered, an error message is displayed saying “ERROR! Please choose a valid option”
* The user chooses option 1, which is to view the calculated individual GPA for any course
* The list of registered course codes, and course names are displayed to choose from
* The user is prompted to enter the course code they want to calculate GPA for
* The course code is validated, in the situation where incorrect course code is entered an error message is displayed saying: “Wrong input entered. Please enter a valid option”
* The course code, and course name are printed, followed by GPA, and the grade obtained by the student in that particular course
* The user chooses option 2, which is to view the cumulative GPA for the given student ID
* The statement “Displaying cumulative GPA for the following courses: “ is printed
* The list of registered courses are printed as their course codes and name
* The cumulative GPA is calculated, and displayed as “Semester GPA: [GPA obtained]”
* The user is asked to choose among the following options to proceed further: to continue with the same ID, to logout the current ID, or to exit the program

**4.5.3 Functional Requirements**

REQ-5.1: Calculation of any individual course GPA

REQ-5.2: The user will be asked to enter the course code to calculate GPA for

REQ-5.3: GPA for that individual course will be calculated and displayed

REQ-5.4: The corresponding grade will be displayed

REQ-5.5: Calculation of cumulative GPA of all registered courses

REQ-5.6: Overall GPA is calculated, and displayed

It can be anticipated that the user enters the incorrect option when choosing among the calculation of individual or cumulative GPA, an error message should be displayed and the user shall be asked to enter the option again. Additionally, the wrong course code can be entered when asked to enter the course code to calculate individual GPA for. The user should be asked to enter the course code again, along with a message informing that the code entered previously was incorrect.

**4.6 Calculate Fee Structure**

**4.6.1 Description and Priority**

The fee structure function displays the total semester fees for any particular student by adding the credit hours of all the registered courses and multiplying by a fixed per credit hour fees, which in this case is 8500.

It then displays the total credit hours for a particular student, the fee per credit hour, followed by the total fees for the semester.

**4.6.2 Stimulus/Response Sequences**

* The user chooses option 5, which is to calculate the fee structure for the semester
* The total number of credit hours for that semester are displayed, as sum of the credit hours of all the registered courses
* The fee per credit hour is displayed, which is set as 8500 Rupees in this instance
* The total fee amount is calculated as total credit hours multiplied by the fee per credit hour, and the obtained amount is displayed
* The user is asked to choose among the following options to proceed further: to continue with the same ID, to logout the current ID, or to exit the program

**4.6.3 Functional Requirements**

REQ-6.1: The user can view the semester fees for their registered courses

This feature does not require any input from the user.

# Other Nonfunctional Requirements

## Performance Requirements

The maximum number of students at the same time can access the program and view the different features back to back, without the program lagging or slowing down. The response time is fast, and the user experience will be smooth, quickly prompting another set of options as soon as the previous request is completed. The system is reliable, even in the instances when a series of incorrect outputs are entered, the program will display error messages, and not crash.

## Safety Requirements

The system will not collect personal data of the user, interfere with user files and other applications running in the background. The authentication feature will ensure that no other user uses the ID and password of any other student. In case of any wrong inputs, relevant error messages will be displayed to help users identify the mistake when prompted to enter.

## Security Requirements

To confer to the security requirements, the first and foremost feature of the Student Management System is for users to authenticate themselves. The data of different users will not be shared across different IDs except for the user themself.

## Software Quality Attributes

* Adaptability: The program has the tendency to be modified according to purpose over time
* Availability: The features of the system are available whenever needed to be used
* Maintainability: The coding of the software is done in a systematic manner, making it easy to understand. Any changes or modifications can be made easily without wasting a considerable amount of time in understanding the logic of the code.
* Portability: No differences in performance will be noted when run on different devices/systems.
* Reliability: All the calculations made are accurate, and functions are available when needed.
* Reusability: Separation of features into various functions makes it easier to reuse the code
* Testability: code is easy to test and identify errors from
* Usability: The system is easy to use, and displays relevant messages whenever an error is made.

## Business Rules

## Only users who have authenticated themselves are allowed to use the program. Moreover only the user themself can view the features of system such as attendance, marks and GPA

**Appendix A: Glossary**

The following acronyms and abbreviations have been used in this document:

IDE: Integrated Development Environment

REQ: Requirement

GPA: Grade Point Average

JDK: Java Development Kit

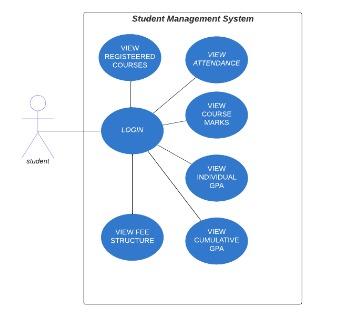
OS: Operating System

**Appendix B: Analysis Models**

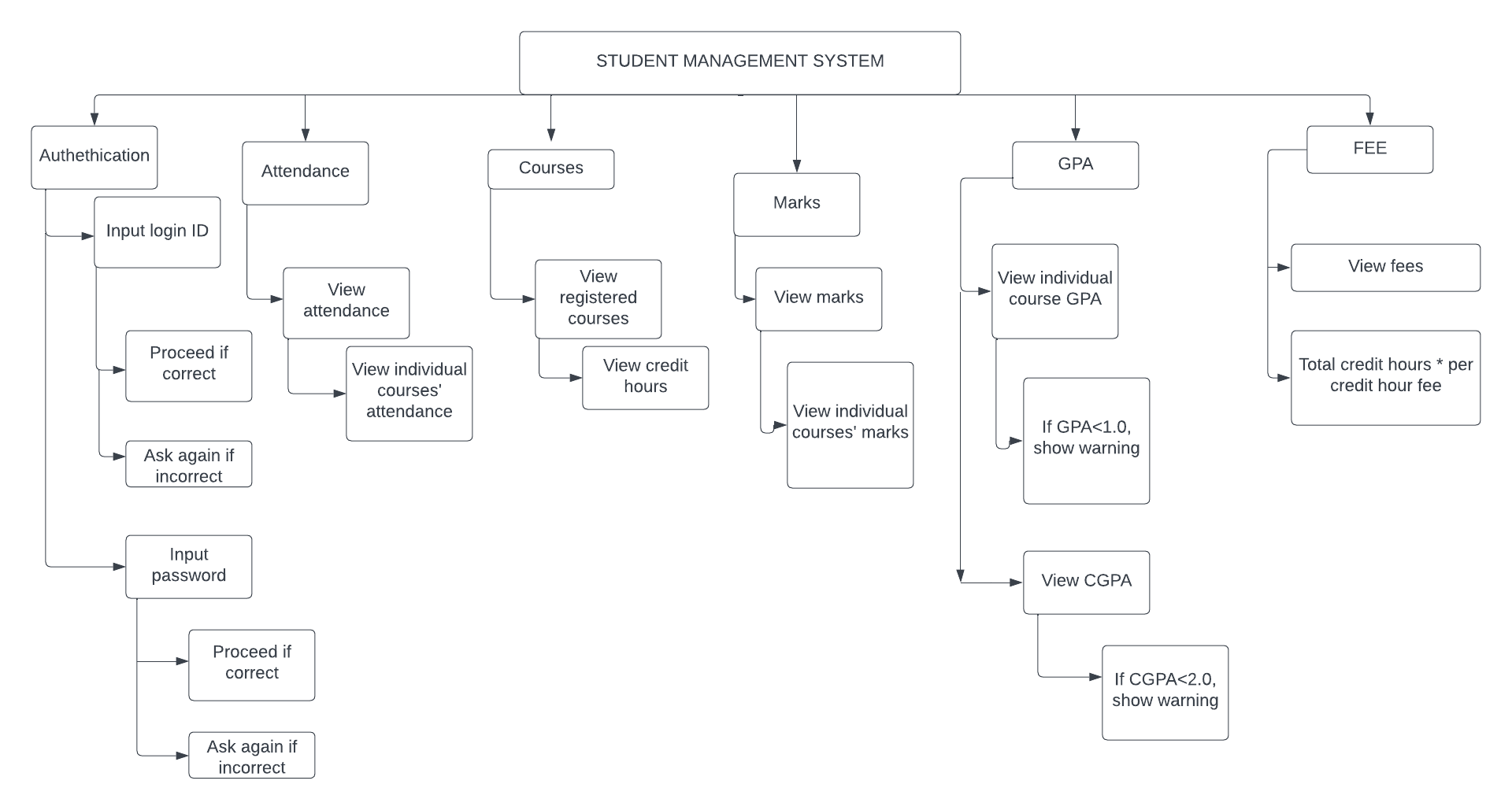
* Class Diagram

| **Student** |
| --- |
| * studentID: String * studentname: String * password: int * studentinfo[ ][ ]: String |
| * setcourseinfo(object): void * authenticationpassword(object, int): boolean * displayregisteredcourses(object): void * displayattendance(object): void * viewcoursemarks(object): void * individualGPA(object): void * cumulativeGPA(object): void * feestructure(object): void * afterfunc( ): int * mainmenu(object): int * setPassword(int): void * getPassword( ): int |

* Use Case Diagram



* Work Breakdown Structure (WBS)



* Activity Diagram

