Software Requirements Specification (SRS)

CSE3001

Software Engineering

(J Component)

Academic Performance Tracker

Prepared by,

SUDARSUN S, 20BCE1699

DEEPAVARSHINI T, 20BCE1661

MADHUMITHAA RP, 20BCE1648

Date: 04th October, 2021

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for the Academic Performance Tracker app. The intended audience of this document includes the admin of the webpage, the content creators, and the end users of the Webapp. The first release of the Accademic Performance Tracker Webapp is projected to have the version number 1.0. This document is to describe its functionality. It is also intended to provide guidance to the requirements team, requirements analyst, design team, and other members of the developing organization.

1.2 Document Conventions

This template is fetched at https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc. Defined terms are highlighted with underlines. Requirements will come with priority to indicate in which order they will be implemented. Versions might be released with only some of the total list of requirements implemented.

1.3 Intended Audience and Reading Suggestions

This document is written for software developers and team mates of the project. Section 2 of the SRS describes the product in more detail. Section 3 provides a complete list of the functional requirements of the intended system. Section 4 provides the non-functional requirements. Section 5 shows the class diagram, and Section 6 the use of case diagram. The document should be read in the order as it is mentioned in the Table of Contents.

1.4 Product Scope

This webapp is being developed for students to track their academic performances. This web app helps students to analyze their level of performance in academics by comparing the present scores with the past achievements. It allows them to predict how hard they should focus on academics so that they can accomplish their targets effectively. It also allows them to track their performance across various activities like CAT exams, Digital Assignments, semester GPAs, etc. They can view their previous scores and search the particular content they want. This helps them to boast their average scores and maintain consistency in the academics so that they can achieve their goals in more effective manner.

1.5 References

https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document- SRS reference

2. Overall Description

2.1 Product Perspective

Our Performance Tracker is a web-based system. This web-app performs basic operations like calculating average marks, storing each input for later use, comparing with previous history, providing inference based on the comparisons, etc. The system provides a secure environment for uploading and storing the academic information and privacy will be maintained for each student.

2.2 Product Functions

- It must have a login page that accepts valid username and password and if the user is new to the system, it must have a sign up page for creating a new account.
- The web app should be able to setup the workspace that helps to create and manage the schedules, assignments, set goals and organize the resources.
- It must be able to get input from the student like marks/grades, tagets to be reached, deadlines for the assignments, total marks of the respective activity, etc.
- The web app must be able to fetch the required information demanded by the student for referring the past academic performances.
- It should be able to store or update the scores given by the student.
- It should also suggest the areas to be improved so that the student can reach his specified target.
- It should be able to produce detailed inference that provides statistical and descriptive analysis based on the average scores.

2.3 User Classes and characteristics

There are 2 main groups of users namely the student and the admin. There should be a common login page through which each can login. If they don’t have an account, they should be able to create a new one for further operations.

The student should be able to input academic particulars, store all data in the database, search desired data for reference and view the detailed report of his/her academic

performance until recent data provided. Hence, students can understand their positives and weaknesses.

The admin will have access over each and every process happening in the system at any stage.

2.4 Design and Implementation Constraints

One of the biggest constraints that will be faced is the student may enter wrong data without knowledge. For example, if the total mark is 100 and the student has entered more than 100, then the data entered is the wrong one. Likewise there can be many possibilities of inaccuracies. Hence, the admin should be able to change the incorrect data to correct one at any time if the student requests to do so.

2.5 User Documentation

All sorts of documents will be provided to the students when they create their account. For any sort of further clarification, they will have to contact the administrator. They can also refer to youtube for tutorial videos.

2.6 Assumptions and Dependencies

The developers assume that all the users who use this application considered being only students. This application is applicable only for students of VIT University. They also assume that the information given for the setup of student workspace is initially correct without any discrepancies.

3. External Interface Requirements

3.1 User Interfaces

The web pages that appear on the screen are built using Python Django Web Framework, HTML and then designed using CSS. The webpages will be interactive with students and be agile for various tasks.

Our Academic Tracker application should contain the following user interfaces,

- Login page for authenticating registered users. This screen should accept user id, password and authenticate against corporate authentication system. It also provides features for "New user registration" and “Forgot password” and "Forgot user id".
- If the student creates a new account, the system should accept various inputs such as schedules, assignments, exam timetable, goals/targets, resources, grades/scores along with highest possible total, etc to setup the student profile and the workspace.

- The home page should be displayed where a registered student can search for academic records based on given parameters. Student can search by course/subject name and fall/winter semester. Search should support handy features such as type-ahead, synonym support, categorized results grouping and suggesting spell correction.
- Search results page displays the results of search operation. The results should fetch and display all the academic records of the specified course in the given semester.
- In the home page, the student can add or update the existing academic data when required. Also, if the student wishes to view the detailed inference of academic history, the system should be able to generate the report and display systematically.
- The interface for the admin will be more flexible so that all the information will be transparent.
- System should be able to log out if the student wishes so.

3.2 Hardware Interfaces

- There is no direct hardware interface specifically for this Academic Tracker application.
- This application can run on any system that has a fully functional web browser preinstalled in it.
- The server will always be on and the user should easily be able to access system
- All the records will be stored in the back end database.

3.3 Software Interfaces

Our Performance Tracker should integrate with the following interfaces,

- The data will be stored in excel database for effective accessibility
- The web app must be able to display all academic data as per the student's pre-requisite which is stored in the server when the student searches for the particular information.
- The system should be always active that the data should be able to be updated at real time and the integration can be done using web services like Django web framework.

3.4 Communications Interfaces

There are no specific communication interface requirements.
Existing OS and network infrastructure will be leveraged for communication.
A full-fledged browser (google chrome preferred) should be used to use this webapp.

4. System Features

4.1 Analysis report of scores:

4.1.1 Description and priority

The user will enter the marks of a particular exam. This score is related to the targets set for the exam and a detailed analysis report of the same is shown to user.

4.1.2 Stimulus/ Response Sequences

The user enters the marks of an exam. This score is compared with the targets set for the particular subject and exam, compares with the previous exam, etc to provide a detailed report on the progress made by the user in terms of marks change and percentage change. The user is made to set new targets for the upcoming exams and provides the average change in performance in comparison to the previous exams.

4.1.3 Functional Requirements:

REQ-1: The data entered by the user will be collected in the database.
REQ-2: The marks are compared with the previous exam to find out the change in marks either increase or decrease.
REQ-3: The marks change is shown to the user.
REQ-4: If there is no marks change, then it shows the user a message “A little progress each day adds up to big results”
REQ-5: If there is no previous exam marks or target marks to compare with, then calculate the average to show the user.

4.2 To do list with reminder

4.2.1 Description and priority

The user can add to do activities to the list with due date mentioned and a can setup an reminder to do the activity.

4.2.2 Stimulus/ Response sequence

The user has to add an activity to the to-do list and mention the due date. The user can then choose to set up a reminder to do the activity. The user can also prioritise the to do list.

4.2.3 Functional Requirements:

REQ-1: The user has to create a to do list.
REQ-2: The user has to enter the due date for the activity.
REQ-3: The user can choose to set up a reminder for activity.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- There should not have any delays to make the system interactive.
- There should be maximum of 2 secs delay in data storage and verification process.
- Uploading and Opening of files, the performance of the application should be less than 2 seconds.
- The dis-connectivity or problem in connecting to the server should be resolved in less than 20 seconds, or the error notice should pop up to notify the user.
- Conversion of file types or uploading of larger files/documents can take up to 10seconds maximum.

5.2 Safety Requirements

Each given by the student must be encrypted so that only if valid login credentials are entered, the data will be decrypted back and displayed normally.

5.3 Security Requirements

Since the data provided by each student is confidential, login is the most necessary requirement. All the data stored in this tracker is allowed to be used by the respective student only. Also, no one is allowed to duplicate our project using the source code.. Since the information in the database is private, strict security implementations must be followed.

5.4 Software Quality Attributes

5.4.1 Reliability

As the system provides the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.

5.4.2 Availability

If the internet service gets disrupted while sending information to the server, the information can be sent again for verification.

5.4.3 Security

The main security concern is for users account hence proper login mechanism should be used to avoid hacking. The student id registration is way to spam check for increasing the security. Hence, security is provided from unwanted use of recognition software.

5.4.4 Usability

As the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

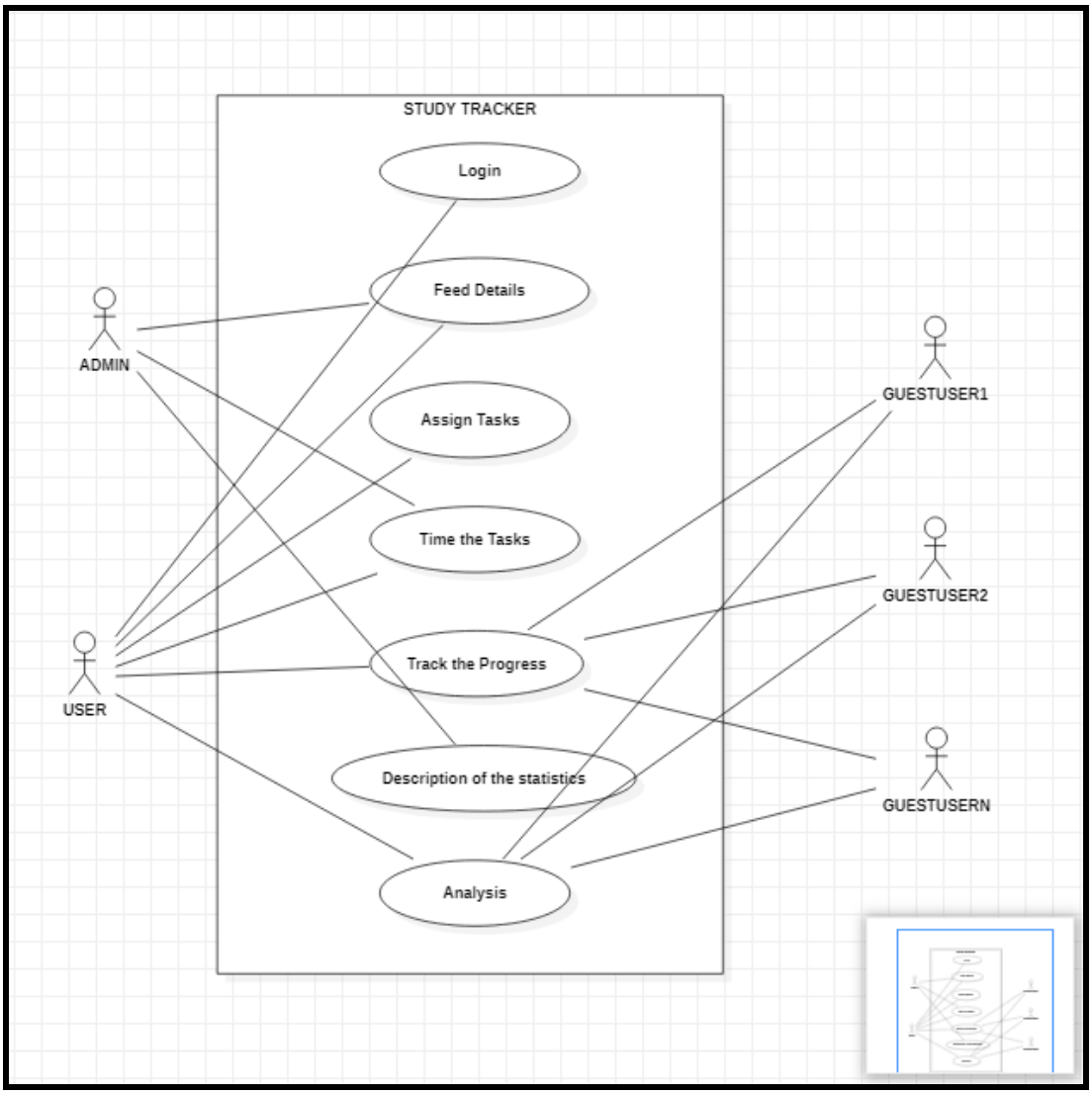
6. Other Requirements

Appendix A: Glossary

Academic Performance Tracker	To analyze student's academic activities
SRS	Software requirement specification.
Student	A person who inputs academic particulars and analyze his/her performances throughout various time periods.
Admin	A person who only has the access to all the information provided by students for proper management of system.

Appendix B:

Use Case diagram:



Appendix C:
To be decided: