
Software Requirements Specification

for

Student Database Management System

Version 1.0

Prepared by

Group #: 26

Group Name: *<team name,
if any>*

Sumit Jaiswal
Sourabh Kulhari
Kartik Rajiv Nirmal
Yuvraj Chandrawanshi

180796
180778
180346
180899

sumitj@iitk.ac.in
sourabhk@iitk.ac.in
kartikn@iitk.ac.in
uvraj@iitk.ac.in

Course: CS253

Mentor TA: *<place mentor TA's name here>*

Date: *<place the date of submission here>*

CONTENTS	II
REVISIONS	II
1 INTRODUCTION	1
1.1 DOCUMENT PURPOSE	1
1.2 PRODUCT SCOPE	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW	1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	1
1.5 DOCUMENT CONVENTIONS	1
1.6 REFERENCES AND ACKNOWLEDGMENTS	2
2 OVERALL DESCRIPTION	2
2.1 PRODUCT OVERVIEW	2
2.2 PRODUCT FUNCTIONALITY	3
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS	3
2.4 ASSUMPTIONS AND DEPENDENCIES	3
3 SPECIFIC REQUIREMENTS	4
3.1 EXTERNAL INTERFACE REQUIREMENTS	4
3.2 FUNCTIONAL REQUIREMENTS	4
3.3 USE CASE MODEL	5
4 OTHER NON-FUNCTIONAL REQUIREMENTS	6
4.1 PERFORMANCE REQUIREMENTS	6
4.2 SAFETY AND SECURITY REQUIREMENTS	6
4.3 SOFTWARE QUALITY ATTRIBUTES	6
5 OTHER REQUIREMENTS	7
APPENDIX A – DATA DICTIONARY	8
APPENDIX B - GROUP LOG	9

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
V 1.0	Full Name	Initial Preparation	16/03/20

1 Introduction

The college administration is prepared to operate as a registration form. Firstly and for admission with the important information about students and retrieve that information to correct it. Student Database Management System deals with all the activities done by computer such as registration and admission process, staff and class management etc.

All these processes are handled by the computer management system. The admission form designs and feeds the information that is required for admission and to maintain the college records.

At the time of registration the Administration should feel the information correctly if any wrong information field in records. Our Software will be used for editing these record lists carefully.

1.1 Document Purpose

The purpose of this document is to present a detailed description of the Student Database Management System. It will explain the purpose and features of the software. It contains detailed functional requirements, non-functional requirements, support requirements and establishes a requirements baseline for development of the system. This document is intended for both the end users and the developers of the software.

1.2 Product Scope

This software system will be a web application wishing to manage the error in students' data online. More specifically to design and develop a simple and intuitive system which will be used for matching the data of each student from the database of csv format with the documents provided by the students during the admission in pdf format. If any error is found, Manager will be able to correct the data in the database.

1.3 Intended Audience and Document Overview

This document is created for,

- The professor of Electrical Department for their review and monitoring the progress of Project.
- The software development team for their use in analysing the requirements.

1.4 Definitions, Acronyms and Abbreviations

- **Admin**

Admin is a user who can be a professor or system administrator and who has the ability to create Managers and TA's and he will assign specific numbers of TA's to each Manager. Admin will distribute the amount of data among managers which have to be checked. Admin can also view as Manager and TA.

- **Manager**

Manager is a user and who has the ability distribute work among the TA's and Manager will be able to edit data in a csv file. Managers can also view as TA.

- **TA**

TA is a user and who has the ability to check in the data provided to him by manager and write a comment corresponding to a particular student if he has found any error in his data. TA will not be able to edit in data.

1.5 Document Conventions

<In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1" margins found in this template. For Section and Subsection titles please follow the template.

TO DO: Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>

The following document conventions are followed in preparing this SRS:

- SRS - Software Requirement Specification
- All keywords related to academics are formatted in *Italics*.

1.6 References and Acknowledgments

The following references are used in preparing this SRS:

- The SRS template sent by Prof. Amey Karkare
- Wikipedia

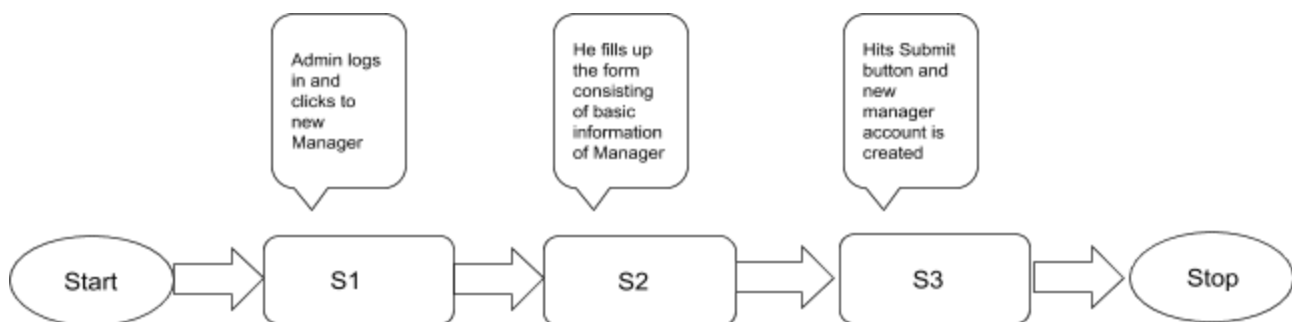
2 Overall Description

2.1 Product Overview

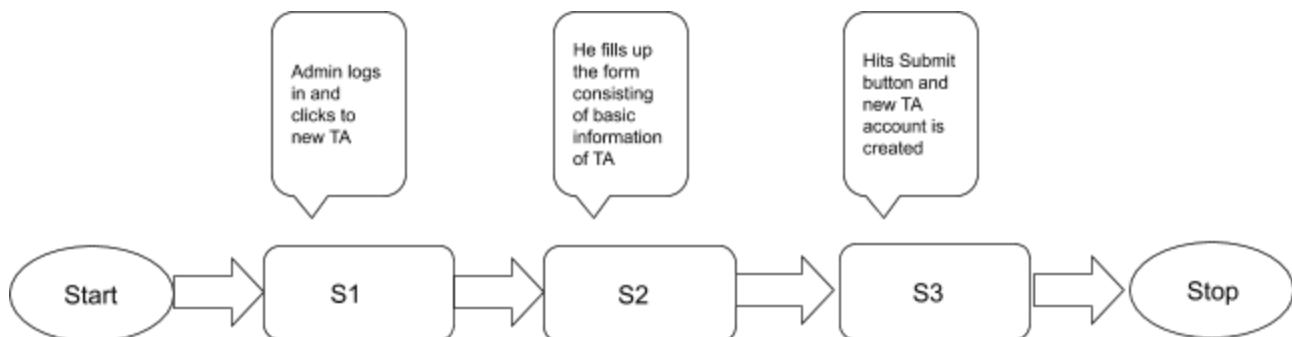
The product Student Database Management system, is an independent product and does not depend on any other product or system. The product will be able to do various tasks associated with correcting student details in a database of csv format.

The following diagrams describe the high level process :

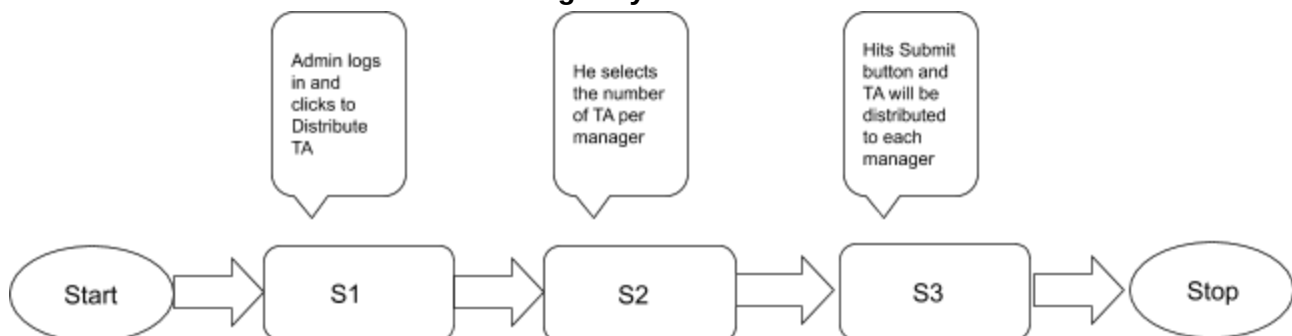
- **Creation of Manager by Admin:**



- **Creation of TA by Admin:**



- **Distribution of TA's under Manager by Admin:**



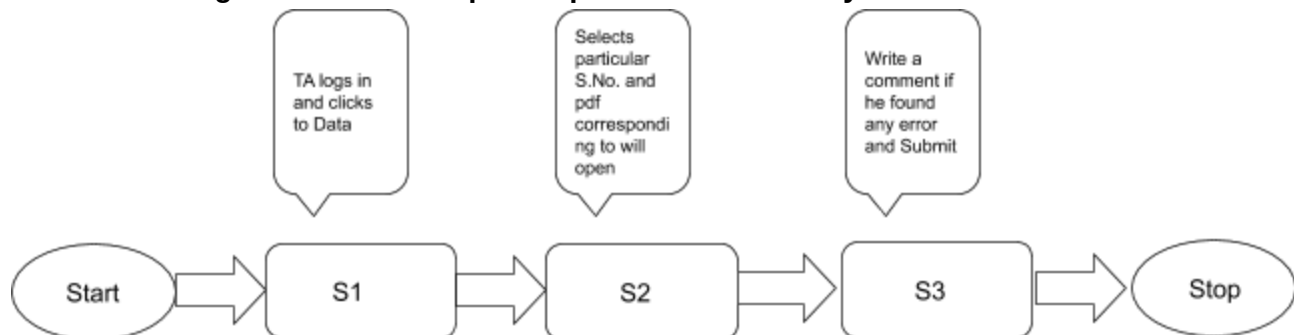
- **Distribution of data by admin:**



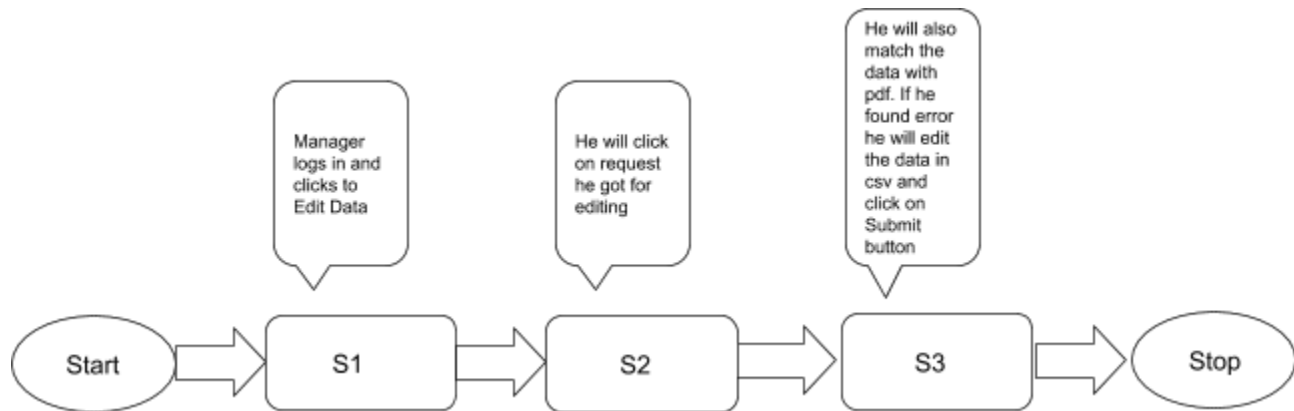
- **Distribution of data by Manager:**



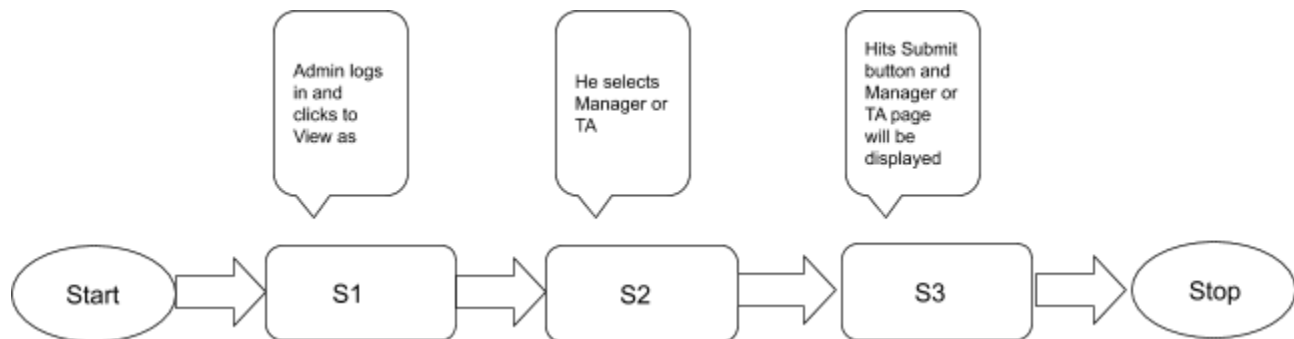
- **Matching data of csv with pdf for particular student by TA:**



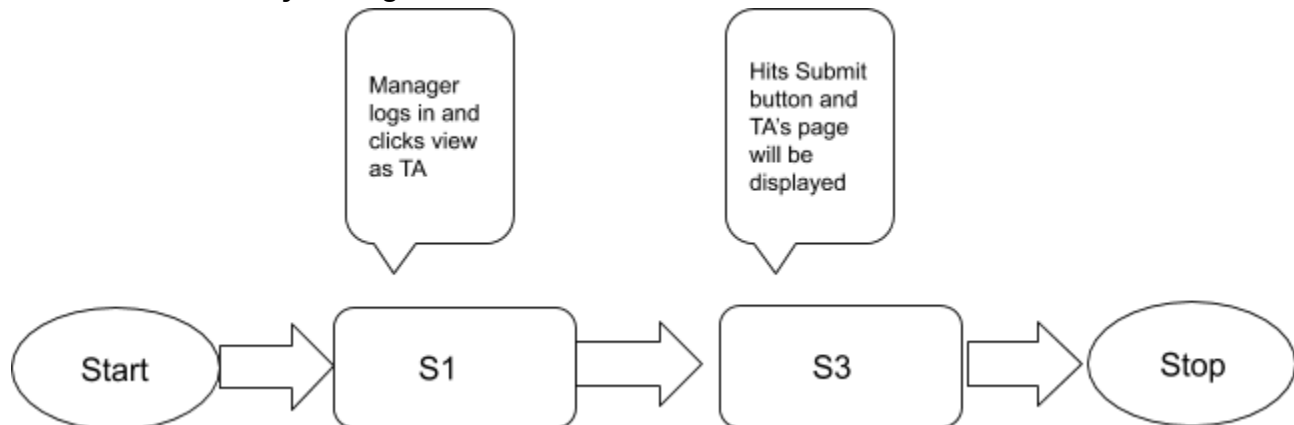
- Editing data in csv by Manager:



- View as Manager or TA by admin:



- View as TA by Manager:



2.2 Product Functionality

This software package is expected to offer following services:

- **For Admin**
Admin is a user who can be a professor or system administrator and who has the ability to create Managers and TA's and he will assign specific numbers of TA's to each Manager. Admin will distribute the amount of data among managers which have to be checked. Admin can also view as Manager and TA.
- **For Manager**
Manager is a user and who has the ability distribute work among the TA's and Manager will be able to edit data in a csv file. Managers can also view as TA.
- **For TA**
TA is a user and who has the ability to check in the data provided to him by manager and write a comment corresponding to a particular student if he has found any error in his data. TA will not be able to edit in data.

2.3 Design and Implementation Constraints

Time constraints:

The software package should be design so as to handle access by 1 Admin , 25 Managers and 250 TAs concurrently.

2.4 Assumptions and Dependencies

There are no assumptions made.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. For your project, you only need to be concerned with the main thermostat (not the mobile app) and can use the graphic from the project description as the basis for your user interface..

TO DO: Provide the graphic for the user interface and provide a basic description as to how users will interact (e.g. menus, etc.).>

3.1.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.

TO DO: Please provide a short description of the different hardware interfaces. This can simply be a list of the devices you must interact with at this point. >

3.1.3 Software Interfaces

<Describe the connections between this product and other specific software components (in your case, just the mobile app that can send commands).>

3.2 Functional Requirements

< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions. >

3.2.1 F1: The system shall ...

3.2.2 <Functional Requirement or Feature #2>

...

3.3 Use Case Model

TO DO: Provide a use case diagram that will encapsulate the entire system and all actors.

3.3.1 Use Case #1 (use case name and unique identifier – e.g. U1)

TO DO: Provide a specification for each use case diagram

Author – Identify team member who wrote this use case

Purpose - What is the basic objective of the use-case. What is it trying to achieve?

Requirements Traceability – Identify all requirements traced to this use case

Priority - What is the priority. Low, Medium, High. Importance of this use case being completed and functioning properly when system is depolyed

Preconditions - Any condition that must be satisfied before the use case begins

Post conditions - The conditions that will be satisfied after the use case successfully completes

Actors – Actors (human, system, devices, etc.) that trigger the use case to execute or provide input to the use case

Extends – If this is an extension use case, identify which use case(s) it extends

Flow of Events

1. Basic Flow - flow of events normally executed in the use-case
2. Alternative Flow - a secondary flow of events due to infrequent conditions
- 3.** Exceptions - Exceptions that may happen during the execution of the use case

Includes (other use case IDs)

Notes/Issues - Any relevant notes or issues that need to be resolved

3.3.2 Use Case #2

...

4 Other Non-functional Requirements

4.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

TODO: Provide performance requirements based on the information you collected from the client/professor. >

4.2 Safety and Security Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.

TODO:

- *Provide safety/security requirements based on your interview with the client - again you may need to be somewhat creative here. At the least, you should have some security for login.*

4.3 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Adaptability, etc...) provide requirements related to the different software quality attributes. Make sure, that you do not just write "This software shall be maintainable..." Indicate how you plan to achieve it, & etc...

*Do not forget to include such attributes as the design for change (e.g. adapting for different degrees, programmes etc.). Please note that you need to include **at least 2** quality attributes. You can Google for examples that may pertain to your system.>*

5 Other Requirements

<This section is **Optional**. Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A – Data Dictionary

<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>