Software Requirements Specification

for

Online College Admission Management System

Version 1.0 approved

Prepared by Vibhu Kumar Singh, Anamaya Vyas, Avnish Tiwari

VIT Vellore

03-04-2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

Traditional college admission is a hectic process, which involves students visiting off-site campus, taking application, filling it and then submission is another hectic story. 0n the day of admission, the flow of candidates is very high and it requires both manual processing and record keeping at the same time that makes the process lengthy and difficult to keep track of the admission status of a candidate in multiple departments. At present admission process is done manually with pen and paper which is very inefficient and utilizes much efforts and time. This college admission management system helps to make the admission process much easier and helps in maintaining database in an efficient way. In this system college admin can add the college details and the stream details. We can get the previous year's cut off marks for all the streams. College can create the cut off list for the current year and the students are expected to register on the website and apply for the desired stream. College can register admissions of new students and also remove the students who denies the admission.

1.1 Purpose

The purpose of this document is to retrieve and analyze the ideas that define the product and requirements that the user needs. This document describes the details of our product, its parameter, and its goals. This SRS document describes the target, audience, user interface of product and Software/Hardware requirements of our product. This document also describes the problem we have faced during the designing and implementation of the product and also describes how we have solved this problem and make our product more efficient.

The management system saves the human power and time cost to perform the same task. The data in the database can be saved for a long time and can be used for different purposes in the future. In management systems, there is a minor chance of losing the data. This document also defines how customers and users see our product and understand the functionality of the product. This document will help the developers/designers in case of maintenance of the software product.

1.2 Document Conventions

Items that are intended to stay in as part of your document are in bold.
Explanatory comments are in italic text.
Plain text is used where you might insert wording about your project.

1.3 Intended Audience and Reading Suggestions

This document is to be read by the development team, the project managers, marketing staff, testers, and documentation writers. The software engineer/Developer and project managers need to become intimately familiar with the SRS. Others involved need to review the document.
Testers need an understanding of the system features to develop meaningful test cases and give useful feedback to the developers. The developers need to know the requirements of the software product they need to build.

This document is for general discussions on the implementation decisions regarding the College Management System. The user of the product should have the concepts of RDMS, SQL, interfaces, and classes.

1.4 Product Scope

- Helping the students in selecting the college.
- Making an exam calendar for the students to keep track of all the exams and not forget about an exam.
- Containing the details about different colleges like placements, amenities, cutoff etc.
- Helps the students in taking admission by providing the links of the college and application form.

Out of Scope-

- Cannot get selected into the college directly.
- Cannot recommend a suitable college for a student.

1.5 References

- https://shsu-ir.tdl.org/shsu-ir/bitstream/handle/20.500.11875/1164/0781.pdf?sequence=1
- https://ieeexplore.ieee.org/document/6208293/
- https://ieeexplore.ieee.org/document/4679917/

2. Overall Description

2.1 Product Perspective

The web pages (HTML) are present to provide the user interface on user(Students and College) interface side. Communication between users and server is provided through HTTP/HTTPS protocols.
The Client Software is to provide the user interface on system user client side and for this TCP/IP protocols are used.
On the server side web server is EJB and database server is for storing the information.

2.2 Product Functions

Some of the features are identified for the software. They are listed below:

- **College Details updation**: The college can update their details after giving the details once any details that they give like fees , average CTC etc is bound to change in near future, which they can update.
- **Data encryption:** Data protection and security is one of the most important aspect of the current times for saving every account's details(Student and College) we encrypt the passwords and usernames so that hacking into system becomes impossible.
- **Email verification:** Who can verify yourself other than you. Keeping that in mind the product will have an email verification system, Forgot password and registration would require email verification.
- **Sort:** Every Student has his own requirements and constraint. Keeping that in mind the product will ask the students the different parameters on which he wants to find a college, the system will sort the data accordingly and give you the result.
- **Fraud protection:** E-Frauds are quite common these days and to protect our customers from this we made a report feature too, in this the customers can report any info or account/user and the administration will verify the reports made by the customers

2.3 User Classes and Characteristics

- The Student and College Admins should have the basic idea to operate (use) the system and he already has the experience to work in the internet (browser).
- Default Language is English.

Some of the users identified for this system through use case analysis are listed below:

- Students
- Data entry operators
- College Administrators
- Product Administrators

2.4 Operating Environment

The CMS is expected to be deployed in a real environment to manage the DBMS inside the college. The centralized database is used to store the information. The user only within the college (members of college staff) can use this management system. Users outside form the college cannot access the management system. This application is developed for windows operating system that can be run on Windows XP and above.
The database is used in different departments within a branch of the college. The database used to store the information is the centralized database. The software we have developed will be installed on different computer systems within a college and software will be connected to a centralized database through LAN within a college and then the user can interact with the system and can store the data and other users can get access the stored through a centralized database.

2.5 Design and Implementation Constraints

Some of the design and implementation constraints identified are listed below:

- Students are not allowed to enter any data entry from College interface.
- Neither Student Nor College has any rights to edit any data in the system.v
- Student can only report an account but its removal is totally lies in the hands of the administration.

2.6 User Documentation

- Online documentation facility is available for the students to assess them for the easy use.
- A specific document should be prepared for the maintenance of the system and should say the system in easiest way.

2.7 Assumptions and Dependencies

- College details fields are already added
- Databases are already created
- Data encryption is already present
- Administrator is already created.

3. External Interface Requirements

3.1 User Interfaces

The user interface for this system will have to be simple and clear. Most importantly, the Information must be easy to read, easy to understand and accessible. The color scheme should be appropriate to provide familiarity there should be no contrast
issues.
The interface would have to be super easy to use and be accessible as well so that the students do not face any problems in navigating through the website.

3.2 Hardware Interfaces

- Intel Core 5(i5-10400)
- Frequency base 2.9 GHZ
- RAM 512 MB or more

3.3 Software Interfaces

- Operating system-We have chosen Windows operating system for its best support and user-friendliness.
- Database-To save the students and college information we have chosen SQL database.
- Language-To implement the project we have chosen Python language for its robustness and its interactive support.

3.4 Communications Interfaces

- Client on Internet will be using HTTP/HTTPS Protocol.
- Client on intranet will be using TCP/IP protocol.

4. System Features

4.1 Registration

4.1.1 Description and Priority

Every college one the time of the login will get a form for filling key points about their college, these key point can be like average CTC, courses they are providing, location etc. Some of the values can remain same but other tend to vary with time for eg: average CTC, NIRF ranking, courses college is proving. For this the product will give the college an option to update the details after entering them for the first time. It is a very important aspect of our product and on a scale of 1 to 10 the priority of this feature is 9.

4.1.2 Stimulus/Response Sequences

After filling the details for the first time the data will be visible on the student dashboard and the student can surf and sort data as per will, the college dashboard will also the data on its own home page and a small button of updating at the bottom which will allow it to direct itself to the updating page which will allow the college interface to update data as per will. After updation, the data will be changed on database and changes will be reflected on the user interface accordingly.

4.1.3 Functional Requirements

REQ-1: Data updation
REQ-2: Details filling
REQ-3: Data verification

4.1.4 Non-Functional Requirements

REQ-1: Data storage
REQ-2: Data security
REQ-3: HTTPS requests generated

4.2 Sorting

4.2.1 Description and Priority

Every student will be able to sort the colleges as per will and the requirements of their own, for eg a student might want a college with less fee, closer location or both. This feature allows the student to compare different colleges with very precise information at just one click. This is one of the salient features of our product and it's a very high priority feature. On a scale of 1 to 10 its priority is 10.

4.2.2 Stimulus/Response Sequences

Complex SQL queries will be used in this features so the stimulus might be a little delayed as data will be searched to provide the best results and with a large database it might take 5-10 seconds. After the sorting the data suitable to the query and requirements will be visible on the user interface which will allow the student to check and compare all the colleges as per will and which suites him the most.

4.2.3 Functional Requirements

REQ-1: Data Sorting
REQ-2: List of sorted data visible in a organized manner
REQ-3: Interface to fill user's choices
REQ-4: Multiple requirements taken care off

4.2.4 Non-Functional Requirements

REQ-1: SQL query usage
REQ-2: HTTPS protocol for contacting the server
REQ-3: Data Scrutiny

4.3 Report

4.3.1 Description and Priority

E-Frauds are one of the most common things these days and if a student thinks that some college is providing fraud data the can report the college by themselves and later the product administrative committee can take the appropriate actions thus we can be sure that no fraud takes place. As The colleges these days are very careful for their reputation and the monitoring of data will be done while registration this feature's priority is not that much. On a scale of 1-10 it can be around 5.

4.3.2 Stimulus/Response Sequences

The report will be saved in the database and the administration will be able to check the problems posted by the students and they can take appropriate action. The results of the action will be visible few days after the complain after scrutinizing the issues posted by the user.

4.3.3 Functional Requirements

REQ-1: Filing problem
REQ-2: Stating the issue

4.3.4 Non-Functional Requirements

REQ-1: SQL query usage
REQ-2: HTTPS protocol for contacting the server
REQ-3: Data Scrutiny

4.4 Admin

4.4.1 Description and Priority

This interface is for the security purposes The admin interface will be hidden from the other users and only few special personals would be able to open it. The admin will be able to see all the users enrolled with the product He will also have the power to remove anyone from the database and can do if he find anything fraudulent. The admin would also settle report queries and take appropriate action accordingly. On a scale of 1-10 it can be around 10.

4.4.2 Stimulus/Response Sequences

The Admin would be able to see all people registered with the website and can delete them and can settle report queries. The admin would be able to contact the database's core and can work with database. He will be using HTTPS queries for the same.

4.4.3 Functional Requirements

REQ-1: View data
REQ-2: View Users
REQ-3: Delete Users
REQ-4: View Reports

4.4.4 Non-Functional Requirements

REQ-1: SQL query usage
REQ-2: HTTPS protocol for contacting the server
REQ-3: Data Scrutiny

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Some Performance requirements identified is listed below:

- The database shall be able to accommodate a minimum of 10,000 records of students.
- The software shall support use of multiple users at a time.

There are no other specific performance requirements that will affect development.

5.2 Security Requirements

Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below. Specific requirements in this area could include the need to:

- Utilize certain cryptographic techniques
- Keep specific log or history data sets
- Assign certain functions to different modules
- Restrict communications between some areas of the program
- Check data integrity for critical variables
- Later version of the software will incorporate encryption techniques in the user/license authentication process.
- The software will include an error tracking log that will help the user understand what error occurred when the application crashed along with suggestions on how to prevent the error from occurring again.
- Communication needs to be restricted when the application is validating the user or license. (i.e., using https).

5.3 Software Quality Attributes

There are a number of attributes of software that can serve as requirements. It is important that required attributes by specified so that their achievement can be objectively verified. The following items provide a partial list of examples.

- The input system will allow for inputting numbers, operands, special symbols and letters of the alphabet.

6. Other Requirements

<NONE> as of now.

Appendix A: Glossary

College Details – A listing of the colleges on the website and the information about them like the facilities, connectivity, courses offered, seat matrix and fee structure. This would help the students in selecting the colleges.

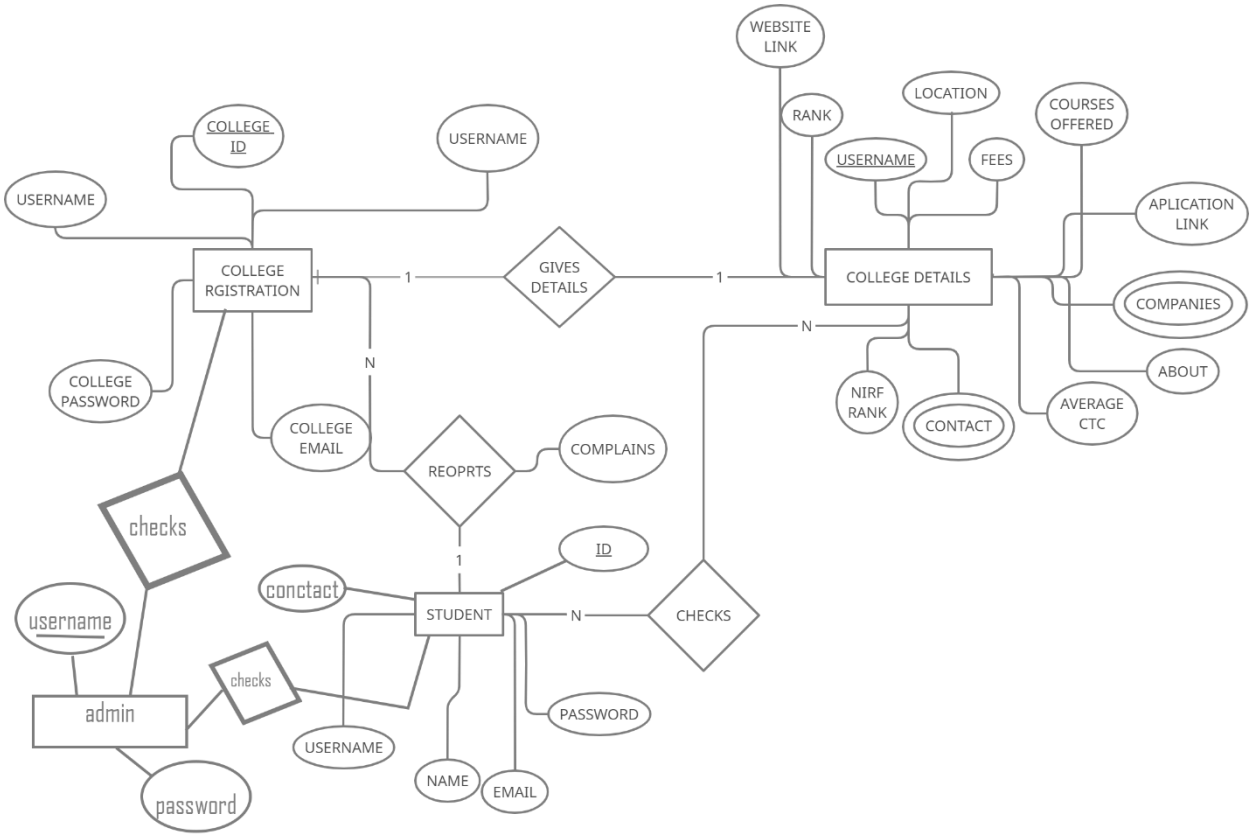
Cutoff-In this section we will list some previous years cutoff of an exam or of college so that students can see which colleges would they be eligible for and they can prepare for their colleges by setting a target.

Exams List-In this we would like to make a calendar like something so that the student can remember all of his exams its due date and its application form filling dates etc. so that he doesn't miss any of the exams.

Student Details-In this section the student would fill his/her details and login and access through site, also if any college wants to contact the student or vice versa the details would help and the students as well as the college would benefit from it.

Appendix B: Analysis Models

ER Diagram:



Appendix C: To Be Determined List

As a team, we will update and evaluate our SRS document every week as we make changes in our design and requirements. We will add new detailed information which will include: research, references, charts and graphs, and more specifications and requirements that we find along the way in the designing and implementation of the product.