Software Requirements Specification

for

RDS++

**Version 2.0 approved**

**Prepared by Sadman Sakib Khan Promit**

**Thinkers**

**21/11/2020**

**Table of Contents**

**Table of Contents [ii](#_heading=h.gjdgxs)**

**Revision History [ii](#_heading=h.30j0zll)**

**1. Introduction [1](#_heading=h.1fob9te)**

1.1 Purpose [1](#_heading=h.3znysh7)

1.2 Document Conventions [1](#_heading=h.2et92p0)

1.3 Intended Audience and Reading Suggestions [1](#_heading=h.tyjcwt)

1.4 Product Scope [1](#_heading=h.3dy6vkm)

1.5 References [1](#_heading=h.1t3h5sf)

**2. Overall Description [2](#_heading=h.4d34og8)**

2.1 Product Perspective [2](#_heading=h.2s8eyo1)

2.2 Product Functions [2](#_heading=h.17dp8vu)

2.3 User Classes and Characteristics [2](#_heading=h.3rdcrjn)

2.4 Operating Environment [2](#_heading=h.26in1rg)

2.5 Design and Implementation Constraints [2](#_heading=h.lnxbz9)

2.6 User Documentation [2](#_heading=h.35nkun2)

2.7 Assumptions and Dependencies [3](#_heading=h.1ksv4uv)

**3. External Interface Requirements [3](#_heading=h.44sinio)**

3.1 User Interfaces [3](#_heading=h.2jxsxqh)

3.2 Hardware Interfaces [3](#_heading=h.z337ya)

3.3 Software Interfaces [3](#_heading=h.3j2qqm3)

3.4 Communications Interfaces [3](#_heading=h.1y810tw)

**4. System Features [4](#_heading=h.4i7ojhp)**

4.1 System Feature 1 [4](#_heading=h.2xcytpi)

4.2 System Feature 2 (and so on) [4](#_heading=h.1ci93xb)

**5. Other Nonfunctional Requirements [4](#_heading=h.3whwml4)**

5.1 Performance Requirements [4](#_heading=h.2bn6wsx)

5.2 Safety Requirements [5](#_heading=h.qsh70q)

5.3 Security Requirements [5](#_heading=h.3as4poj)

5.4 Software Quality Attributes [5](#_heading=h.1pxezwc)

5.5 Business Rules [5](#_heading=h.49x2ik5)

**6. Other Requirements [5](#_heading=h.2p2csry)**

**Appendix A: Glossary [5](#_heading=h.147n2zr)**

**Appendix B: Analysis Models [5](#_heading=h.3o7alnk)**

**Appendix C: To Be Determined List [6](#_heading=h.23ckvvd)**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Version2 | 23/11/20 | Version1 was incomplete. | 2 |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to present a detailed description of the RDS++. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document will be proposed to the honorable faculty for its approval.

## Document Conventions

Default conventions are used.

## Intended Audience and Reading Suggestions

This document is intended for the developers of the system and the honorable faculty.

## Product Scope

This software system will be a University Portal system with student monitoring feature. The system will have student profiles’ details, all their grades history, attendance history. The faculties can take attendance manually in the system. The system can monitor the students through the webcam and network traffic and take attendance automatically and also detect cheating.

## References

IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

# Overall Description

## Product Perspective

The system RDS++ consists of a university web portal and a browser plugin/extension. The portal will be interacted by students, faculties and admin. The plugin can read browser network traffics and also access the webcam to monitor the students. This management system is controlled by the teachers and system administrators. In this system, admin is the main user who has full access to the management system. Admin can view and modify all information that is stored in the database.

Admin can view and modify the student’s records like student’s profile, attendance, fee, results, and details of teachers and other employees in college, their personal information and their attendance for their salaries.

Teachers have access to view and modify the student’s information like their attendance, marks of exams to generate the progress report of students. When the teacher “update’s the student’s information then admin can view this information.

## Product Functions

The functional requirements of this system are:

* Register new students.
* Record the attendance of students.
* Record the internal marks of students.
* Record the feed details of students.
* Register a new teacher/employee.
* Register a new user for the system.
* Record the salary details of employees.
* Record the course details and subject information.
* Record the scholarship details and information.
* Detect cheating during exam.
* Give attendance automatically if rules are followed.
* Generate various reports for all transactions in the system.

## User Classes and Characteristics

The RDS++ contains the following key features:

• Admin:

Admission /Registration: This will deal with all new students to register them with their personal information/details.

Staff: Admin will add new staff/Teachers/Employees by inserting their personal information and register them.

Admin can add a new user for the RDS++ by assigning them a login/username and password

Finance: Admin can view and manage the financial transaction for the University.

Rules: Admin can modify cheating and attendance rules.

Scholarship: Admin can assign scholarships to the students.

• Employee:

Personal information: personal details of staff (name, address, contact, etc.) will be stored.

Attendance: store and represent the attendance details of staff.

Salary: provides details about the salary of staff

Student information: Teachers can view the details of their students and provide access to the teacher to “update” and manage the student’s exam marks.

• Student:

Personal information: This will provide all the details about a student like his/her name, address, Guardian, etc.

Attendance: This feature provides the attendance details of the student.

Marks: This will provide the marks of student internal marks in exams.

Fee: This will use to keep a record of student fees and in the future by using this feature Admin can view the fee record of any student within the college.

Department:

This will provide the details about departments within a college with their name and every department have its Department name.

Course:

This will provide the details about courses that are offering by the college.

Every course has a name and its unique name. Every course has different subjects and every subject has its own unique name.

* Faculty:

Faculty can take students attendance manually.

Faculty can grade students.

Faculty can allow certain websites that can be visited by the students without getting flagged.

* System:

Generate student’s profile information.

Detect cheating.

Take attendance automatically.

## Operating Environment

Operating environment for the RDS++ system is as listed below.

* *Operating system:* Windows 7, Windows 8, Windows 8.1, Windows 10 or later, OS X Yosemite 10.10 or later, 64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+, Android 5.0+ (Lollipop), iOS 12.2 or later
* *Browser*: Google Chrome for full support/Any for portal support
* *CPU*: Intel Pentium 4 processor or later that's SSE3 capable
* *Device*: Computer, Mobile, Tablet, any with an internet browser.
* *Ram*: 128MB+
* *HDD*: 100MB+
* *Webcam*: Mandatory only for full support.

## Design and Implementation Constraints

Relational Database Design and Implementation: Clearly Explained, Fourth Edition, provides the conceptual and practical information necessary to develop a database design and management scheme that ensures data accuracy and user satisfaction while optimizing performance. Database systems underlie the large majority of business information systems. Most of those in use today are based on the relational data model, a way of representing data and data relationships using only two-dimensional tables. This book covers relational database theory as well as providing a solid introduction to SQL, the international standard for the relational database data manipulation language.

Constraints:

1. Financial problems
2. Lack of knowledge among users in using the system
3. Lack of product developers
4. Collection of user’s actual details is difficult
5. The AI isn’t perfect yet.

• Concepts needed to create effective relational database designs.

• Theoretical and practical guidelines for normalizing relational databases.

• The impact of database design on data accuracy and consistency.

• Examples of how design can inhibit or boost database application performance.

• Using SQL to implement database structure, manipulate data values, and retrieve data.

• The overall systems analysis and design process during which database design takes place.

• Database implementation issues including database security, concurrency control, and integration with XML.

• Techniques for using CASE tools for database design.

• The relationship between relational database design and data warehouse design

## User Documentation

* HTML (Hyper Text Markup Language): It is used to create static web pages.
* CSS: Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.
* Python: Python is an interpreted, high-level and general-purpose programming language.
* Django: Django is a Python-based free and open-source web framework that follows the model-template-views architectural pattern.
* HTTP (Hyper Text Transfer Protocol): It is a transaction-oriented client/ server protocol between a web browser and a web server.
* XML (Extensible Markup Language): It is a markup language that was designed to transport and store data.
* MySQL: It is a database management system that provides a flexible and efficient database platform to raise a strong "on demand" business applications.
* OpenCV: OpenCV is a library of programming functions mainly aimed at real-time computer vision.
* JavaScript: JavaScript is high-level, often just-in-time compiled, and multi-paradigm.

## Assumptions and Dependencies

* There exists the relationship between Role and PIM is one to many.
* There exists the relationship between Person and Authentication is one to many.
* There exists the relationship between Family details and Person is one to many.
* There exists the relationship between Person and Experience is one to many.
* There exists the relationship between Person and Qualification is one to many.
* There exists the relationship between Person and Address is many to one.

Functions of actors of the system:

Faculty:

* Can conduct test and tutorials
* Can view student profile
* Can post forums
* Can view and post feedback
* Can generate test reports
* Can view student report

Student:

* Can select faculty according to their interested field ace
* Can take up test and view their test reports
* Can view mentor
* Can post forums/feedbacks

Administrator:

* Can view student/faculty status
* Can analysis new student/faculty request
* Can cancel or delete student/faculty request

Localization: English

Overview: The SRS include two main sections, namely,

User Interfaces: A user interface is the view of a database interface that is seen by the user. User interfaces are often graphical or at least partly (GUI-graphical user interface) constructed and offer tools which make the interaction with the database easier.

* Hardware Interfaces:

The server is directly connected to the client system. Also, the client has the access to the database for accounting details and storing the login time. The client access to the database in theserver is read only.

Processor minimum: Intel Pentium 2 processor Memory (RAM): 256 MB DDR Hard disk: 4GB Display: 1024\*786, Keyboard: Any normal keyboard

Product Functions:

Login capabilities: The system shall provide the users with logon capabilities.

• Alert:

 The system can alert in case of any problem.

* The system shall allow the users to access the system from Internet using HTML. It uses web browser interface at client side.

Software Interfaces: E- Mentoring for women is a multiuser multitasking environment. It enables the user to interact with the server and attain access to the internet and also leaves a record in the inbuilt database. It uses java servlets a front-end programming tool and SQL as the backend application tool.

* Operating System: Windows 9x/xp, Windows ME.
* Database: MySQL Server. Application: ASP (Active Server Pages) Java Virtual Machine.
* Front End Client: Online Ticketing system uses the front end as Java, Java Server Pages and HTML at the client side
* Advanced Tools: Net beans 6.7.1 and JDK.
* Web server: Apache Tomcat is used as the web server for the Online Ticketing System.
* Data Base Server: MySQL server is used as the database server for Online Ticketing.
* Back End: Online Ticketing system uses MySQL Ver 14.12 Distrait 5.0.75 as a back-end database.

Communications Interfaces

Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

Specific Requirements: Use Case Administrator Responsible for managing all the two types of users, viewing logs and managing standard groups of the system.

Manage System users: The Administrator will provide the system mentor and mentee the authentication to use the site.

View Logs: Responsible for checking the logs of different system user for auditing and maintaining the integrity of the system.

System Reports: The Administrator is responsible to generate the system reports for the future reference.

View All Details: View the user’s details, forum details and mark records.

Back Up Date: The Administrator is responsible to back up all the data at a particular time every day.

Manage System Users:

Name of the Use Case: View Systems users

Description: View the list of system users in the role and view the details of roles, tasks and permissions assigned to the users.

Preconditions:

Administrator is already logged in.

Users have already been created and assigned some roles, tasks and permissions.

Normal Flow of Events:

The system user or the role will be selected.

Query will be submitted.

Relevant output will be displayed.

Alternate Flow of Events: None

Post Condition: None

3. Supplementary Requirements: Have Hours of Operation that are 24\*7: Because System Can be authenticated it can stay open for 24 hours a day. Make the existing website more Dynamic in nature: Many early web implementations consisted of static HTML pages. This because it becomes Difficult to manage if the number of pages get too large. Application should serve dynamic user based customized web pages to its client from server. Tie the existing website into the existing enterprise system: Any existing website that relies from the manual duplication of data from another system is one that can be improved.

# External Interface Requirements

## User Interfaces

Will be similar to NSU RDS system.

## Hardware Interfaces

Will be added later.

## Software Interfaces

Will be added later.

## Communications Interfaces

HTTP is used.

# System Features

Will be added later.

## System Feature 1

Will be added later.

## System Feature 2 (and so on)

Will be added later.

# Other Nonfunctional Requirements

Will be added later.

## Performance Requirements

All pages should be loaded within three seconds or less, assuming a broadband connection on the client side. Therefore, response time for transactions will be three seconds or less. The system should support as many as 50 online users simultaneously with negligible response delay.

## Safety Requirements

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.

## Security Requirements

Only authorized users will be permitted to access the system. The system will provide additional security. The system will provide the users with a secure way to change their passwords, whether when initializing a new account or by user request. If requested, we will upgrade the system to use the HTTPS protocol in subsequent iterations in order to prevent unauthorized third-party viewing of the contents.

## Software Quality Attributes

Availability: The Student Course Registration System will be available to registered students, faculty and staff 99.9% of the time unless otherwise noted.

Correctness: The SCRS will operate correctly from user log in and should reach the checkout page when ready to register.

Maintainability: System admin shall maintain and correct student catalog information to ensure teachers, schedules, classes are current and up to date

Usability: SCRS shall satisfy all users and meet their needs.

The quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database

## Business Rules

All users need a valid email address. An email is considered valid if the user can prove they have access to the email address associated with the user account. The enrollment system shall include a mandatory email input on the user enrollment form page. If the Submit button is pressed on the user enrollment form page and all mandatory fields are present, the enrollment system shall send a confirmation email to the email address provided. If the acknowledgement link within theconfirmation email is clicked, the enrollment system shall create a new user account using confirmed enrollment details.

# Other Requirements

Will be added later.

**Appendix A: Glossary**

Will be added later.

**Appendix B: Analysis Models**

Will be added later.

**Appendix C: To Be Determined List**

Will be added later.