Software Requirements Specification

for

<ONLINE RATING SYSTEM>

Version 1.0 approved

Prepared by <RAJAT SINHA>

<NIIT UNIVERSITY >

<17 SEPT 2016>

Table of Contents

Table of Contents ii

Revision History 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 2

2.6 User Documentation 2

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 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of the document is to collect all ideas that have come up to describe the online rating system. Basically we are covering the services provided by NIIT UNIVERSITY. This will help to provide an overview of our software i.e. online rating system.

## Document Conventions

SRS – SOFTWARE REQUIREMENT SPECIFICATION

UML – UNIFIED MODELLING LANGUAGE

## OP – ORIGINAL POSTER

## Intended Audience and Reading Suggestions

People who are involved in development of the product can add more features related to the product.

## Product Scope

SRS is also aimed at specifying requirements of software to be developed. Our software will aim at collecting the reviews about

University facilities like mess, gym, tuck shop, etc. and send the report to higher entity so that necessary action can be taken by the desired authorities. Students using our software can obtain solutions regarding to their problems.

Finally after we took a survey from the students of our University, we got mostly positive responses with few negative response regarding the size of the audience.

The most important suggestion is that the ratings should be reviewed from time to time and if rating is below average, the service should be looked into as soon as possible.

## References

1) This project is under the guidance of Professor Amit Kumar, CSE Department,NIIT University.

# Overall Description

## Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

## Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

## User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## 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, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## 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.>

# System Features

The System features are divided into two parts between student and report worthy authorities.

The complaints would be handled on our system software and would be available on anyone’s domain who would like to check.

**4.1 Login and Register (actors – student & authority)**

**4.1.1 Description and priority**

Authentication and authorization would be the two main features. These are very important as security would be highly dependent on these features.

**4.1.2 Stimulus / Response Sequences**

**4.1.3 Functional Requirement**

**4.2 Review for Rating the Complain (actor­ - student)**

**4.2.1 Description and priority**

With this feature, students would be able to give feedback to the authorities whether the complaint resolved was up to the mark or not.

**4.2.2 Stimulus / Response Sequences**

**4.2.3 Functional Requirement**

4.3 up vote, comment

**4.3.1 Description and priority**

If two or more students have the same complaint, then it would be better if there is a method of commenting otherwise the list of complaints would be too long and it would be a burden for the authorities to address all.

**4.3.2 Stimulus / Response Sequences**

**4.3.3 Functional Requirement**

**4.4 Complaint Records (actors-student, authority)**

**4.4.1 Description and priority**

Other users will also have access to see the complaints of students to the authorities after they login and register.  
**4.4.2 Stimulus / Response Sequences**

**4.4.3 Functional Requiremen****ts**

**5. Other Nonfunctional Requirements**

5.1 Performance Requirements

• Application Services and Technical support - Programmers can sort out all the bugs and errors through accessing source code. Network admin will have to maintain 24-hour uptime.

5.2 Safety Requirements

Backup data storage – There will be a backup facility to address in case of server failure.

5.3 Security Requirements

• Firewall & Waterwall - Unneccesary users and entities will be blocked from accessing the service through the use of firewall and waterwall.

• Administration Features – Each student can access only his/her registration records. Admin has the full right to access all student’s registration records.

5.4 Software Quality Attributes

• Bug/ERROR report feature – Whenever there is bug/error, report will be sent to the development team for further improvement.

6. Other Requirements

6.1 Hardware Requirements

• Network – The university would be having access to the internet whether it would be LAN or Wifi.

• Server – A server would be required for user access.

• Client Computers – Mac and Windows client computers, etc.

• Production support systems –Related hardware support (back-up tapes, UPS) etc

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>