**NOTE: This template is shareware downloaded from** [**www.processimpact.com**](http://www.processimpact.com)**. All shareware payments are donated to the Norm Kerth Benefit Fund to help a consultant who is disabled with a brain injury. Please visit** [**http://www.processimpact.com/norm\_kerth.html**](http://www.processimpact.com/norm_kerth.html) **to make a shareware payment ($10 suggested). Thank you!**

Software Requirements Specification

for

Top Five

Version 1.0 approved

Prepared by Beatrice Cerda

UHD

March 18, 2017

eLearning versions of several popular Process Impact training seminars are available at [www.processimpact.com/elearning.shtml](http://www.processimpact.com/elearning.shtml), including “In Search of Excellent Requirements,” “Exploring User Requirements with Use Cases,” “Writing High-Quality Requirements,” “Software Inspections and Peer Reviews,” and “Project Management Best Practices”. Single-user and corporate-wide site licenses are both available.

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 Project Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 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. System Features 3

3.1 System Feature 1 3

3.2 System Feature 2 (and so on) 4

4. External Interface Requirements 4

4.1 User Interfaces 4

4.2 Hardware Interfaces 4

4.3 Software Interfaces 4

4.4 Communications Interfaces 4

5. Other Nonfunctional Requirements 5

5.1 Performance Requirements 5

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 6

Appendix C: Issues List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Beatrice Cerda | 03/18/17 | Initial Draft | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

This project is to create a Student Information Management System that helps a university IT department improve their services and for management to track student information. The software system only stores and retrieves student’s partial information during the current semester such as student’s name, student ID, courses registered, exam grade, and GPA.

## Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

## Intended Audience and Reading Suggestions

Intended audience, the client.

## Project Scope

The project is designed to assist in managing in an efficient yet simple manner, the entry of student grades and calculation of student GPA. The system is not designed to integrate all aspects of an educative system, but instead provide a bare bones operation enabling multiple administrators data entering their own courses and students.

## References

1. Team Website : <https://www.c9.io>

2. Course Home Page: <https://www.bb.uhd.edu>

# Overall Description

## Product Perspective

This system will consist of two parts, the back end database and the front end website interfaces.

## Product Features

The website will allow faculty to add students, courses, and grades to the system allowing the students to query their GPA. The website will be linked to a database enabling the faculty to add to the database.

## User Classes and Characteristics

Website Class: transfers requests from user to control class.

Admissions Class: controls all movements between all other classes and authenticates users.

Registration Class: enables the faculty to add courses and students, plus serves to verify registration.

Records Class: allows the modification of a record by adding grades to already existing records.

Student Report Class: permits display of student records on screen.

## Operating Environment

Cloud based service with MySQL and PHP capabilities.

## Design and Implementation Constraints

Time

Resources

Man-power

## User Documentation

Online Tutorials of MySQL and PHP

## Assumptions and Dependencies

Team of 5 resources.

Software Availability

# System Features

<<<insert use case diagram and class diagram and sequence diagram>>>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

3.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

3.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

3.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# External Interface Requirements

## User Interfaces

<<<insert diagram of GUI>>>

<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

No Hardware Interfaces

## Software Interfaces

MySQL database with Microsoft operating systems, Git tools, and PHP libraries.

## Communications Interfaces

Website: PHP and HTML

# Other Nonfunctional Requirements

## Performance Requirements

Not Applicable

## Safety Requirements

Not Applicable

## Security Requirements

Basic Authentication, must practice sound security habits.

## 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.>