**Software Requirements**

**Specification**

**for**

**Enhanced Faculty Loading System**

**Version 1.0 approved**

**Prepared by Ochotorena, Rafael**

**Asia Pacific College**

**November 23, 2016**

**Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document. Software Requirements Specification for Faculty Loading System.**

# Table of Contents

[Table of Contents 2](#_Toc467694377)

[Revision History 2](#_Toc467694378)

[1. Introduction 1](#_Toc467694379)

[1.1 Purpose 1](#_Toc467694380)

[1.2 Document Conventions 1](#_Toc467694381)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc467694382)

[1.4 Product Scope 1](#_Toc467694383)

[1.5 References 1](#_Toc467694384)

[2. Overall Description 1](#_Toc467694385)

[2.1 Product Perspective 1](#_Toc467694386)

[2.2 Product Functions 2](#_Toc467694387)

[2.3 User Classes and Characteristics 2](#_Toc467694388)

[2.4 Operating Environment 2](#_Toc467694389)

[2.5 Design and Implementation Constraints 2](#_Toc467694390)

[2.6 User Documentation 2](#_Toc467694391)

[2.7 Assumptions and Dependencies 2](#_Toc467694392)

[3. External Interface Requirements 3](#_Toc467694393)

[3.1 User Interfaces 3](#_Toc467694394)

[3.2 Hardware Interfaces 3](#_Toc467694395)

[3.3 Software Interfaces 3](#_Toc467694396)

[3.4 Communications Interfaces 3](#_Toc467694397)

[4. System Features 3](#_Toc467694398)

[4.1 System Feature 1 4](#_Toc467694399)

[4.2 System Feature 2 (and so on) 4](#_Toc467694400)

[5. Other Nonfunctional Requirements 4](#_Toc467694401)

[5.1 Performance Requirements 4](#_Toc467694402)

[5.2 Safety Requirements 4](#_Toc467694403)

[5.3 Security Requirements 5](#_Toc467694404)

[5.4 Software Quality Attributes 5](#_Toc467694405)

[5.5 Business Rules 5](#_Toc467694406)

[6. Other Requirements 5](#_Toc467694407)

[Appendix A: Glossary 5](#_Toc467694408)

[Appendix B: Analysis Models 5](#_Toc467694409)

[Appendix C: To Be Determined List 5](#_Toc467694410)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Rafael Ochotorena | 11/26/16 | Document Created | 1.0 |
| Rafael Ochotorena | 11/29/16 | Updates for necessary changes | 1.1 |

# Introduction

## Purpose

This document is version 1.1 of SRS, it describes how the product, Enhanced Faculty Loading System aims how to improve the teaching capability of Asia Pacific College, how the product is used, who uses it and how it was developed. This document also explains the step by step interface of the product and all other requirements the product needs.

`

## Document Conventions

While reading this document, certain words or phrases might be used for specific reasons. The document conventions are the following:

Algorithm

A process or set of rules to be followed in calculation or other problem-solving operations, especially by a computer.

Power BI

It specifically shows how the loading is done and how the criteria are weighted.

Intranet

Since the Flavio System can only be accessed within the school, it is because its network is based on TCP/IP protocols belonging to an organization which is Asia Pacific College and it can only be accessed by the faculty and students within the school area.

## Intended Audience and Reading Suggestions

This document is intended for the Executive Director and the faculty members of the School of Computing and Information Technology in Asia Pacific College. The rest of the Software Requirements Specification document contains the overall description of the product such as its functions and user classes, the external requirements of the product which explains the walkthrough of the product and how the product came to be, and also all other requirements the product needs for its functionality.

## Product Scope

This product aims to quicken the process of faculty loading for the Executive Director by giving her less paper works, for the faculty to have more time preparing their lesson plans, and for the students to receive a better quality of teaching. This product will also use an analytic-based reporting system that provides a comprehensive report on faculty loading. If this product is integrated, it will truly give efficiency for the Director’s workload.

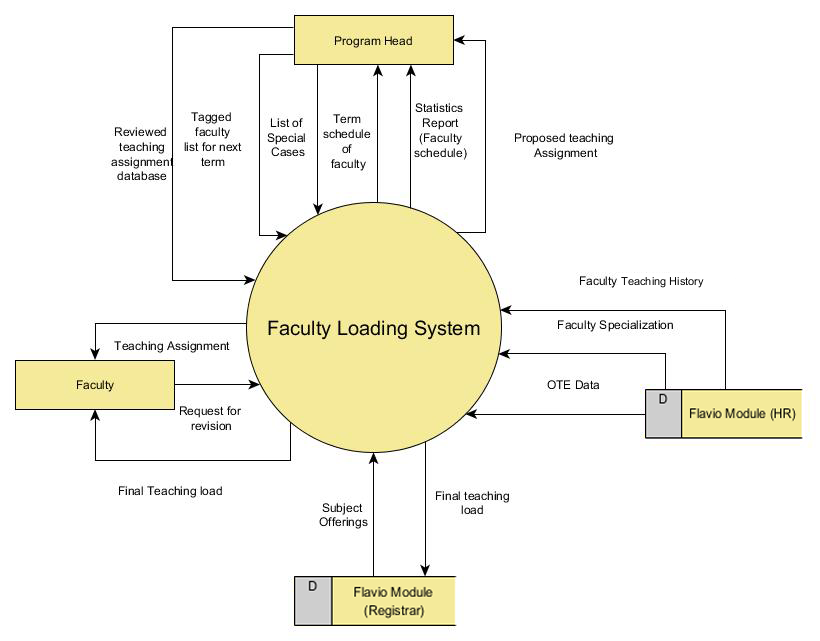
## References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

# Overall Description

## Product Perspective

This product will replace the old manual process of faculty loading with an enhanced faculty loading system that will be efficient and effective. This product is a component of the Flavio system by which all necessary data are received from Flavio.



## Product Functions

The major functions the product are the following:

* Matching of faculty to their respective schedules
* Algorithm for careful considerations of factors used for matching
* Reporting system to provide a report on how the matching is done

## User Classes and Characteristics

This product is aimed the Executive Director and the faculty members but the faculty members can only view on their schedules. The Executive Director, however, will use the system for loading the faculty members’ schedule. Both the users may use the system anytime but the system will be used specifically before the start of the next term.

## Operating Environment

The software would be able to operate to any operating system but Windows 7 (or later) would be the most preferred to use. Any web browser may be used for the system since it will be integrated to the Flavio system of Asia Pacific College.

## Design and Implementation Constraints

As stated in the main document to which the criteria for loading is mentioned, some rules must be followed:

* No 3 consecutive classes for each faculty
* No more than 21 units for full-time faculty
* No more than 15 units for part-time faculty
* No more than 8 hours of work load each day for each faculty
* No more than 4 preparations for each faculty

If the system would be integrated, specific softwares are highly recommended, HTML and PHP and MySQL for the database management. The school will be responsible for maintenance once the system is deployed.

## User Documentation

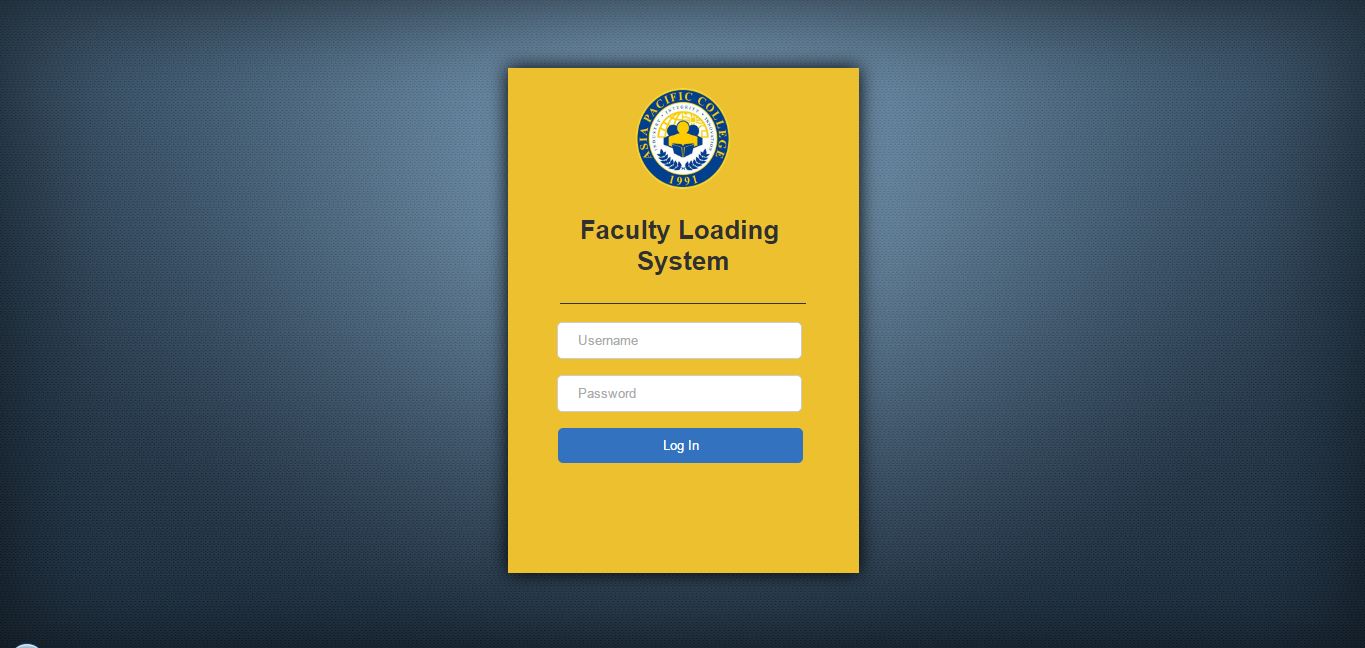
Other than this document, other documentations were also delivered for this product one of which is found in projects2.apc.edu.ph/wiki under CSPROJ2 projects. Also, other documents can also be found on GitHub.

## Assumptions and Dependencies

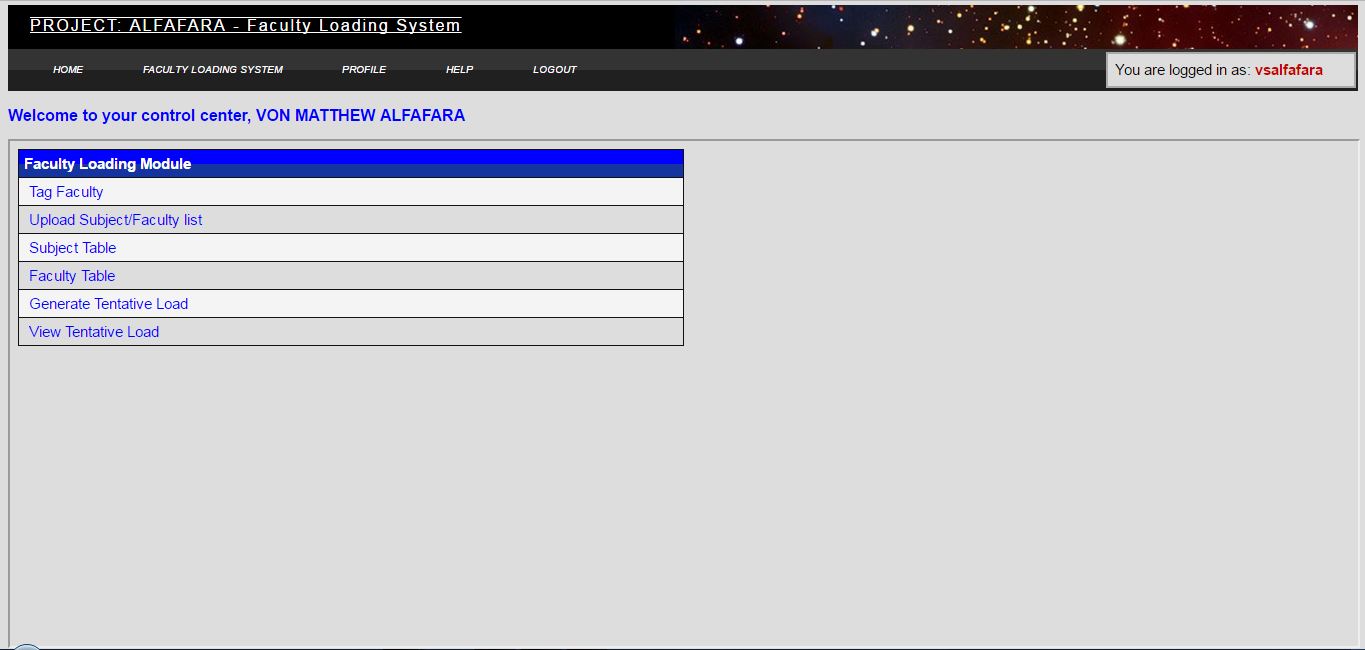
<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions 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

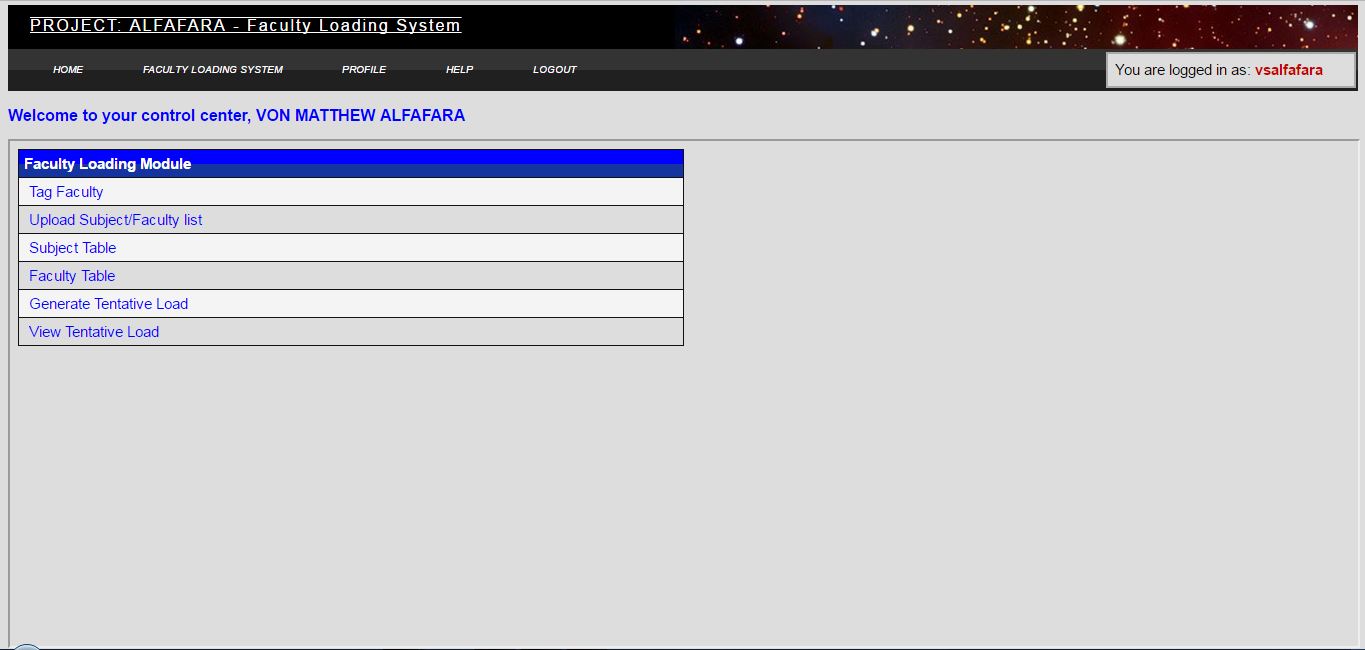
## User Interfaces

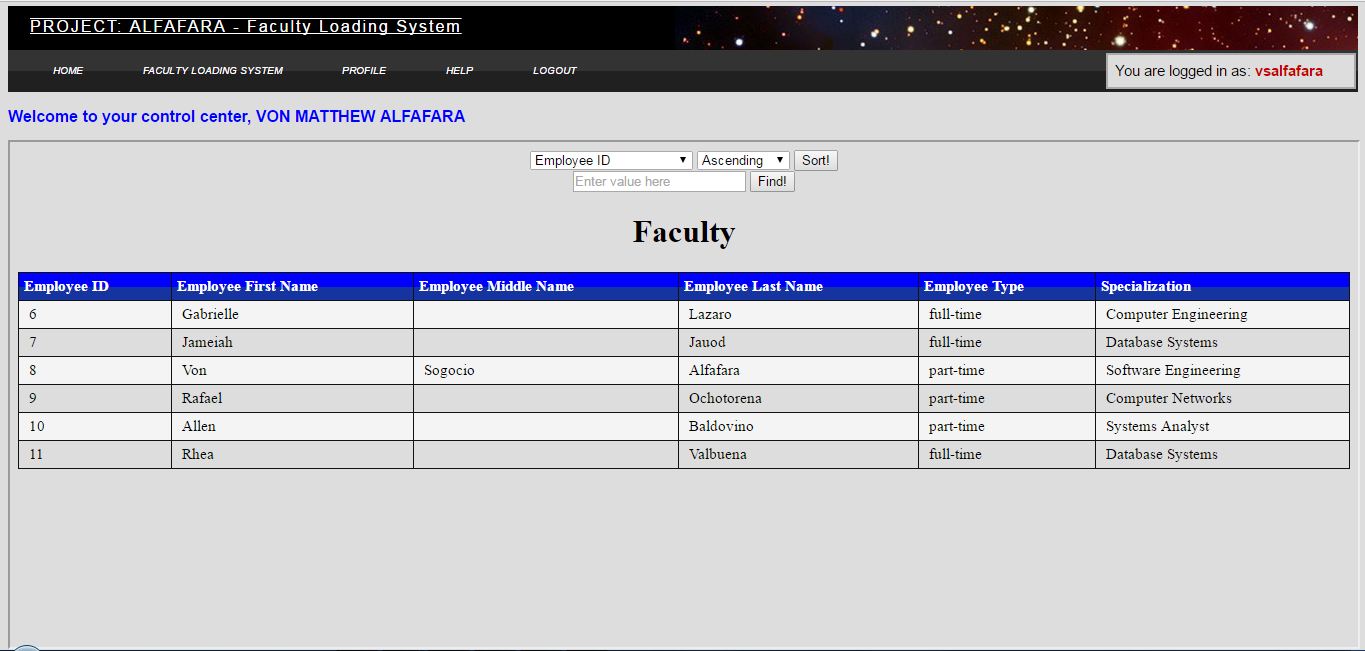
In order for the software to be used, the user must first login to the Flavio System.

If the user is the Executive Director, he/she would be able have access to all software control.

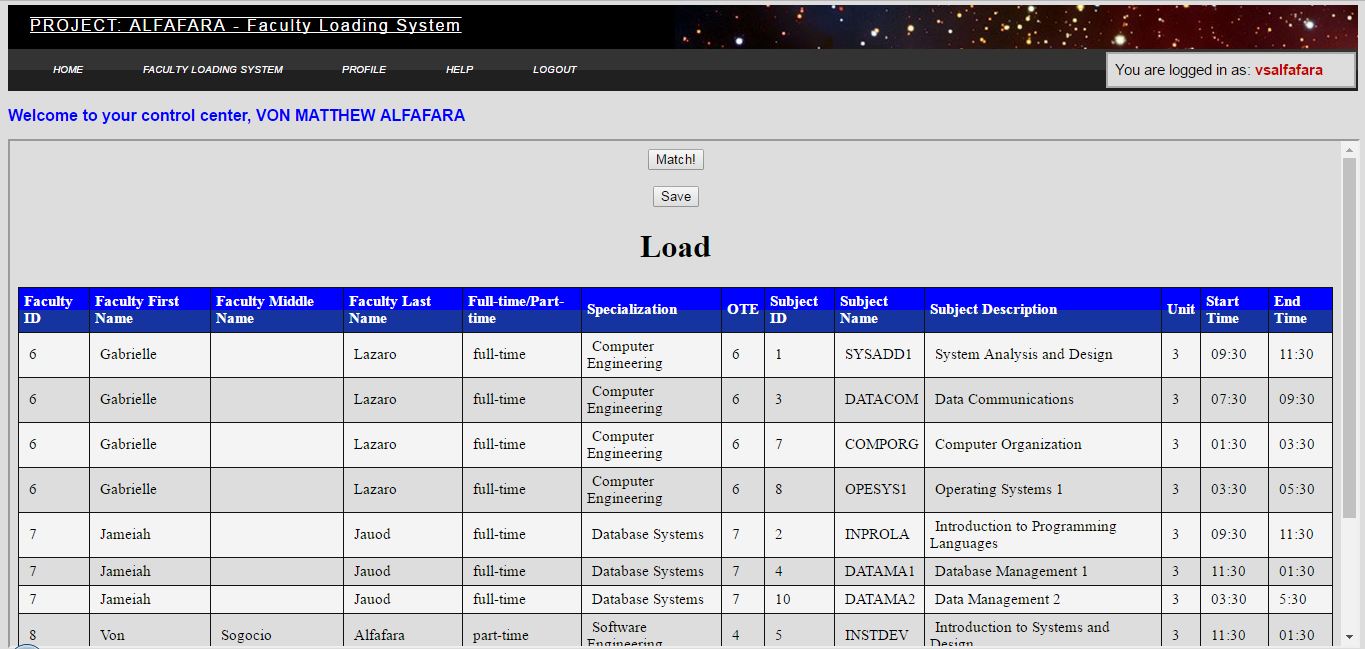


He/she would be able to view the data necessary for loading.





To which he/she would be able to load the schedules of faculty members.



## Hardware Interfaces

The hardware components used for the product is a dual core 2.4 GHz+ processor, an 8GB RAM.

## Software Interfaces

The product was created through a cross-platform source code editor specifically notepad++ and Sublime Text, a cross-platform web server XAMPP, MySQL for the database, HTML, CSS and PHP, and all possible web browsers that a user may use.

## Communications Interfaces

The product is most likely be used by the school, Office365 would be used in means of communication. Google Chrome, Mozilla Firefox, and Microsoft Edge are the most used browsers nowadays to which it might be used by the users of the product. Since the product is integrated to the Flavio System, it could only be accessed within school territory because intranet is used.

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.> 4.1.1 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).>

4.1.2 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.>

4.1.3 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)

# Other Nonfunctional Requirements

## 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.>

## Safety 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.>

## Security Requirements

<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. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## 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.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<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: 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.>