CS360

Software Requirements Specification (SRS) Template

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.

The document in this file is an annotated outline for specifying software requirements, adapted from the IEEE Guide to Software Requirements Specifications (Std 830-1993).

Tailor this to your needs, removing explanatory comments as you go along. Where you decide to omit a section, you might keep the header, but insert a comment saying why you omit the data.

CS360

Team #3

MyIPFWAdvisor

Software Requirements Specification

Document

Version:(n)

Date: 1-24-2012

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this project is to aid IPFW CS students in the scheduling of courses.

1.2 Scope

MyIPFWAdvisor will assist students in scheduling by automating some of the roles of an advisor. The software will make it easier for CS majors to ensure they are on track with scheduling, while keeping a class schedule that fits with their personal and work schedules.

1.3 Definitions, Acronyms, and Abbreviations.

Spring - Java web framework we will be using to develop the user interface.
Google Web Toolkit (GWT) - Another Java-based web framework we have considered.
Spring Roo - A tool to aid in the construction of classes for Spring.
Sugar - CSP Solver software, written in Perl.
Cream - Java API to Sugar.

1.4 References

Frameworks, APIs and Tools may be found at the following websites:
Spring, Spring Roo: http://www.springsource.org
GWT: http://code.google.com/webtoolkit/
Sugar: http://bach.istc.kobe-u.ac.jp/sugar/
Cream: http://bach.istc.kobe-u.ac.jp/cream/

1.5 Overview

This document will overview the requirements for the project.

2. The Overall Description

The software is a scheduling advisor website. There will be a web-based user interface to choose courses with the help of an artificially intelligent software on the server-side.

2.1 Product Perspective

The product will integrate with several existing systems.

2.1.1 System Interfaces

External systems that we must integrate with are the world wide web, IPFW's database systems and the chosen server.

2.1.2 Interfaces

MyIPFWAdvisor will expose a web interface to users (students and advisors).

2.1.3 Hardware Interfaces

The system has no known hardware interface requirements.

2.1.4 Software Interfaces

System has no known Sofware Interface requirements.

2.1.5 Communications Interfaces

HTTP/HTTPS

2.1.6 Memory Constraints

There are no known memory constraints.

2.1.7 Operations

2.1.8 Site Adaptation Requirements

Our system requires a server with a modern Java installation.

2.2 Product Functions

- "Bingo Sheet" maintenance for BS and BA Computer Science majors as well as BS Information Systems majors
- On-demand printing of "Bingo Sheet"

- Highlighting of unfulfilled requirements
- Automatic identification of courses that fulfill each specific course requirement
- Course scheduling profile tailored to the preferences of each student
- Automatic schedule generation based on scheduling profile and available courses
- Recommendations for what course to take to fulfill a specific General Education requirement based on past CS/IS majors' preferences
- Advisor query capabilities to, for example, list all majors who are eligible to take a given course or who have received a given grade in a completed course.

2.3 User Characteristics

Users will fall into one of two categories:
1. University Students
2. University Advisors

In both cases, technical skill may be limited. Therefore, interface should be intuitive for anyone with at least a highschool education, as this is the one characteristic we can guarantee of our users.

2.4 Constraints

1. The system will be managing confidential information, and thus must (at least) conform to university security protocols.
2. In general, the system must integrate with existing course registration systems.

2.5 Assumptions and Dependencies

2.6 Apportioning of Requirements.

3. Specific Requirements

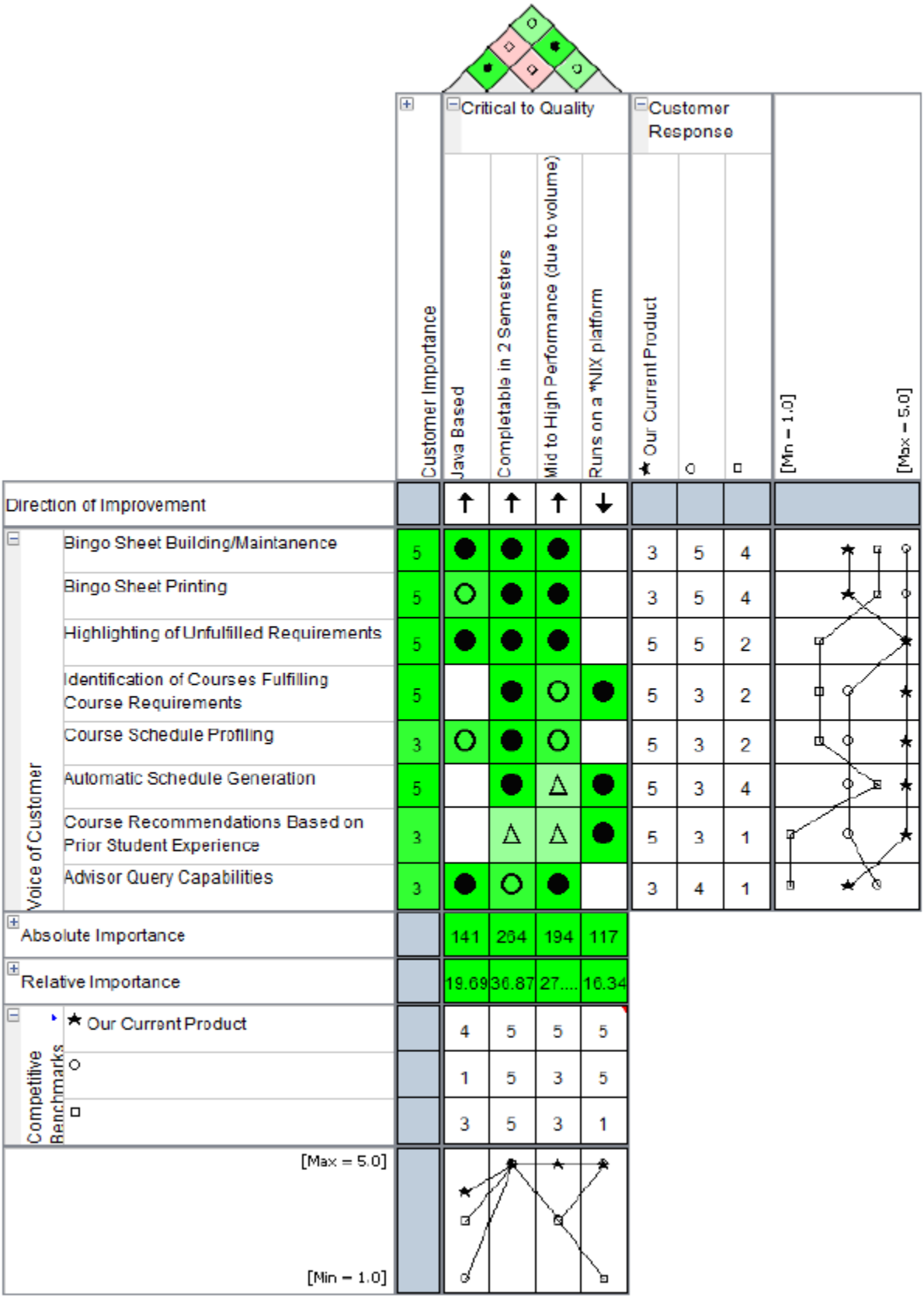
3.1 External Interfaces

1. The system shall interface with existing course registration databases.
2. The system shall interface with either Sugar or Cream.
3. The system shall interface with Spring and/or GWT.

3.2 Functions

FR	MyIPFWAdvisor
FR1	Interface Layer
FR1.1	Shall provide a secure web-based interface
FR1.2	Shall automatically highlight unfulfilled requirements
FR1.3	Shall automatically identify course that fulfilll each requirement
FR1.4	Shall facilitate building and maintaining bingo sheets
FR1.5	Shall provide on-demand bingo sheet printing
FR1.6	Shall facilitate advisor course information queries
FR2	Business layer
FR2.1	Shall facilitate student course advising and scheduling
FR2.1 .1	Shall generate course schedule profiles
FR2.1 .1.1	Shall automatically generate schedule
FR2.1 .1.2	Shall automatically advise courses
FR2.2	Shall access student database and keep track of records
FR3	Infrastructure Layer
FR3.1	Shall store student schedule preferences for later use
FR3.2	Shall store administratively defined bingo sheets
FR3.3	Shall be able to connect to IPFW's LDAP/Databases

3.3 Performance Requirements


Critical to Quality
Customer Response
Customer Importance
Java Based
Completable in 2 Semesters
Mid to High Performance (due to volume)
Runs on a *NIX platform
Our Current Product
[Min = 1.0]
[Max = 5.0]
Direction of Improvement
Voice of Customer
Bingo Sheet Building/Maintanence
Bingo Sheet Printing
Highlighting of Unfulfilled Requirements
Identification of Courses Fulfilling Course Requirements
Course Schedule Profiling
Automatic Schedule Generation
Course Recommendations Based on Prior Student Experience
Advisor Query Capabilities
Absolute Importance
141
204
194
117
Relative Importance
19.69
30.87
27....
16.34
Competitive Benchmarks
Our Current Product
[Max = 5.0]
[Min = 1.0]

3.4 Logical Database Requirements

Any IPFW CS Student or Faculty with a computer and internet connection is capable of using the website, with unlimited frequency of use.

3.5 Design Constraints

3.5.1 Standards Compliance

3.6 Software System Attributes

3.6.1 Reliability

The system must scale to meet the demands of its potential user base.

3.6.2 Availability

The system must have 24/7 availability via its web interface.

3.6.3 Security

1. The system shall use HTTPS for communications.
2. The system shall integrate with existing university security protocols.

3.6.4 Maintainability

1. The system must handle the absence of external systems.

3.6.5 Portability

The software will be host independent to the extent possible with the described requirements.

4. Change Management Process

Changes to this document will be transferred to each group member via email, basecamp, or Google Docs. These changes will be reflected within a new and updated version of the

SRS document.

5. Document Approvals

Trent Forkert, Marat Kurbanov, Connor Becker, Alek Bouillon, YeiSol Woo