

CS460

Software Requirements Specification (SRS)

MyIPFWAdvisor

CS460

Team #3

MyIPFWAdvisor

**Software Requirements Specification
Document**

Version:2

Date: 5-01-2012

Table of Contents

- [1. Introduction](#)
 - [1.1 Purpose](#)
 - [1.2 Scope](#)
 - [1.3 Definitions, Acronyms, and Abbreviations.](#)
 - [1.4 References](#)
 - [1.5 Overview](#)
- [2. The Overall Description](#)
 - [2.1 Product Perspective](#)
 - [2.1.1 System Interfaces](#)
 - [2.1.2 Interfaces](#)
 - [2.1.3 Hardware Interfaces](#)
 - [2.1.4 Software Interfaces](#)
 - [2.1.5 Communications Interfaces](#)
 - [2.1.6 Memory Constraints](#)
 - [2.1.7 Operations](#)
 - [2.1.8 Site Adaptation Requirements](#)
 - [2.2 Product Functions](#)
 - [2.3 User Characteristics](#)
 - [2.4 Constraints](#)
 - [2.5 Assumptions and Dependencies](#)
 - [2.6 Apportioning of Requirements.](#)
- [3. Specific Requirements](#)
 - [3.1 External Interfaces](#)
 - [3.2 Functions](#)
 - [3.3 Performance Requirements](#)
 - [3.4 Logical Database Requirements](#)
 - [3.5 Design Constraints](#)
 - [3.5.1 Standards Compliance](#)
 - [3.6 Software System Attributes](#)
 - [3.6.1 Reliability](#)
 - [3.6.2 Availability](#)
 - [3.6.3 Security](#)
 - [3.6.4 Maintainability](#)
 - [3.6.5 Portability](#)
- [4. Change Management Process](#)
- [5. Document Approvals](#)

1. Introduction

1.1 Purpose

The purpose of this project is to aid IPFW CS students in the scheduling of courses through a semi-automated system in which a Student picks which courses he would like to take, along with a constraints such as the minimum and maximum number of credit hours he is willing to take. The System then comes up with a potential schedule of courses, attempting to match the provided inputs as closely as possible.

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.

This team will not be implementing the actual artificial intelligence, as it has already been implemented. However, we aim to connect the Registrar's database, the AI, and construct a web-based user interface for the underlying system.

1.3 Definitions, Acronyms, and Abbreviations.

Sugar - CSP Solver software, written in Perl. This is used by the AI system to generate the potential schedule..

ODS/Banner - Two names for the Registrar's database.

1.4 References

Frameworks, APIs and Tools may be found at the following websites:

Sugar: <http://bach.istc.kobe-u.ac.jp/sugar/>

iCal4J: http://wiki.modularity.net.au/ical4j/index.php?title=Main_Page

JQuery: <http://api.jquery.com/>

Tomcat: <https://tomcat.apache.org/>

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. The System will attempt to construct a schedule that meets the user-specified requirements.

2.1 Product Perspective

The product shall interface to ODS

The product shall integrate with IPFW's existing user-authentication systems

The product shall approximate the look and feel of new.ipfw.edu

2.1.1 System Interfaces

The product shall interface to ODS

The product shall integrate with IPFW's existing user-authentication systems

The product shall be accessible through a secure, web-based interface.

2.1.2 Interfaces

MyIPFWAdvisor will expose an intuitive web-based 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 Software Interface requirements.

2.1.5 Communications Interfaces

The System shall communicate over the following protocols and systems

HTTP

HTTPS

OracleDB

Postgres

2.1.6 Memory Constraints

There are no known memory constraints.

2.1.7 Operations

2.1.8 Site Adaptation Requirements

The system requires a server with a modern Java installation.

The system requires access to databases within IPFW. For security reasons, this access requires either a VPN connection, or a connection on-site at IPFW.

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” via browser printing capabilities.
- Highlighting of unfulfilled requirements
- Automatic identification of courses that fulfill each specific course requirement on initial login
- Automatic schedule generation based on user selections and available courses
- Ability to view current course schedule
- Advisor capabilities to act on behalf of the student for major actions, and edit the digitized Bingo Sheet.

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 all 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.
3. The system must prevent a user from generating multiple schedules simultaneously.

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 Sugar
3. The system shall interface with JQuery.

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 fulfill 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 implement code-level security measures
FR2.2	Shall implement total encapsulation (POJO/Java Beans)
FR2.3	Shall generate course schedule profiles
FR2.3.1	Shall automatically generate schedules based on the scheduler input
FR2.3.2	Shall automatically advise courses based on the scheduling output
FR3	Infrastructure Layer
FR3.1	Shall store student schedule preferences for later work
FR3.2	Shall store administratively defined bingo sheets for use in scheduling

Software Requirements Specifications Document

FR3.3	Shall be able to connect to IPFW's LDAP/Database structure
FR3.4	May communicate with users and administrators via e-mail
FR3.5	Shall log all events

3.3 Performance Requirements

Software Requirements Specifications Document

						<input type="checkbox"/> Critical to Quality			<input type="checkbox"/> 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/Maintenance	5	●	●	●		3	5	4		
	Bingo Sheet Printing	5	○	●	●		3	5	4		
	Highlighting of Unfulfilled Requirements	5	●	●	●		5	5	2		
	Identification of Courses Fulfilling Course Requirements	5		●	○	●	5	3	2		
	Course Schedule Profiling	3	○	●	○		5	3	2		
	Automatic Schedule Generation	5		●	△	●	5	3	4		
	Course Recommendations Based on Prior Student Experience	3		△	△	●	5	3	1		
	Advisor Query Capabilities	3	●	○	●		3	4	1		
	Absolute Importance			141	204	194	117				
Relative Importance			19.69	30.87	27.00	16.34					
Competitive Benchmarks	★ Our Current Product		4	5	5	5					
	○		1	5	3	5					
	□		3	5	3	1					
[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. The system shall only allow a Student or any Advisors to access said Student's information

3.6.4 Maintainability

1. The system must handle the absence of external systems, even if only by returning an error.

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