

Software Requirements Specification

Report and Analysis of Data

for RIT Office of Cooperative Education and Career Services

Version 1.4
(waiting customer approval)

Prepared by Chris Nurre

NOBY

Monday, February 11, 2008

Table of Contents

Introduction	2
Purpose	2
Document Conventions	2
Intended Audience and Reading Suggestions	2
Project Scope.....	2
References	2
Overall Description.....	2
Product Perspective.....	2
Product Features.....	2
User Classes and Characteristics	2
Operating Environment	2
Design and Implementation Constraints	2
User Documentation	2
Assumptions and Dependencies.....	2
Testing Environment	2
System Features	2
Session Setup Feature Set.....	2
Student Service Feature Set.....	2
Employer Service Feature Set.....	2
Placement Service Feature Set.....	2
Activity Service Feature Set.....	2
External Interface Requirements.....	2
User Interfaces	2
Software Interfaces	2
Communications Interfaces	2
Other Nonfunctional Requirements.....	2
Performance Requirements	2
Security Requirements.....	2
Software Quality Attributes	2
Future Development and Requirements	2
Explanation.....	2
Future System Features	2
Future Services.....	2

Revision History

Name	Date	Reason For Changes	Version
Chris Nurre	02-Feb-08	Initial Version	1 (draft)
Chris Nurre	05-Feb-08	Revised Feature Sets	1.1 (draft)
Chris Nurre	06-Feb-08	Added to missing sections and kiosk service	1.2 (draft)
Chris Nurre	10-Feb-08	Updates per customer comments. Changed 'kiosk service' to 'activity service' Version rejected	1.3
Chris Nurre	11-Feb-08	Updated new parts based on customer comments. Version for customer to approval.	1.4

1. Introduction

1.1 Purpose

The product described by this document will be completed by the NOBY Senior Project Team and will be made for RIT Office of Cooperative Education and Career Services. The primary focus of the project will be on the second tier (in respect of a three tier architecture) and the primary services required for basic querying of the current data sources.

1.2 Document Conventions

Each requirement within a feature set has a designator in the form of:

REQ-*f.r* (e.g: REQ-1.1)

In this form *f* is the number of the feature set, and *r* is the requirement number.

1.3 Intended Audience and Reading Suggestions

The intended audience for this document is the members of the NOBY software team, and the project customers (the O.C.E.C.S). The most important part of this document are the specified feature sets because these will dictate the progress and direction of the project.

1.4 Project Scope

The scope of this product will lie within the second tier of the entire web architecture, its primary services, and the interfaces in and out of the services. The primary services include the Employer, Student, Placement, and Activity services. The interfaces required will be the authentication model and connection to the data sources via a connection pooling mechanism. As the services are completed it will be necessary to create basic web interfaces but any extra interfaces or additional services will be on the responsibility of the R.I.T O.C.E.C.S. Finally some methods on how to potentially increase speed will be investigated and possibly implemented if feasible.

1.5 References

For additional information on architecture of the product see: *ArchitectureDiagram.pdf*

2. Overall Description

2.1 Product Perspective

The purpose of this project is to create a replacement for the current R.I.T O.C.E.C.S reporting system using web services. While the current system encompasses all tiers of a typical web application, this product will focus on the second tier of the system and provide services that allow for a standard interface to retrieve the data already provided. The services will not necessarily

analyze or perform computations against the data retrieved. The responsibility of the official presentation tier will be on the O.C.E.C.S office and other consumers of the services.

2.2 Product Features

The major product features will be the retrieval of the core data(students, employers, placements, and activities) related to the reporting and analysis of co-op and full time job information.

2.3 User Classes and Characteristics

The primary users of the reporting services are the O.C.E.C.S office and R.I.T students. Secondary users of this information include department staff and faculty.(These secondary groups are being actively researched to give more adequate user classes). One common characteristic of the users are that they must be logged in through the R.I.T CAS Authentication service to make use of reporting functions of this product.

2.4 Operating Environment

The web services tier will be hosted on a Windows Server 2003 box, will be written in C# and run under the .NET 2.0 environment, with all data will be hosted on a SQL Server 2000. Calls from outside clients to the services will be made using the SOAP protocol.

Although the current system relies on the above technologies, the system should be built in a way that allows for easy migration to a new environment. The most likely future migration will be a move from SQL Server 2000 to SQL Server 2005.

2.5 Design and Implementation Constraints

The final product must be a self contained Visual Studio 2005 project and use subversion (SVN) for version control. The O.C.E.C.S does not have a set coding standard, but the source should be well commented.

2.6 User Documentation

The final release of the product will include a full documentation of the API. This document will will be created using the NDoc generation tool.

2.7 Assumptions and Dependencies

This product relies that the current database structure will remain the same in the future. Any modifications could have adverse effects and the accuracy of returned reports cannot be guaranteed.

Secondly, the authentication moduel makes use of the R.I.T C.A.S authentication server. Should the interface to this service change, the built-in authentication module will most likely fail.

2.8 Testing Environment

All access points will be testing using and automated testing architecture. This will be used to confirm the accuracy of all returned datasets and to record speed improvements over the existing system.

3. System Features

3.1 Session Setup Feature Set

3.1.1 These features are necessary for the majority of the services below to operate properly. All access points provided by this system will require a logged in client. Many of these requirements are access points allowing for the basic setup of the query pages and supply lookups for selection menus.

3.1.2 Stimulus/Response Sequences

Most of these functions will be used by query pages to setup pull down menus or other selection boxes. Excluding the the first requirement, all access points will be directly called using the SOAP protocol and will result in a SOAP response.

3.2.3 Functional Requirements

REQ-1.1: verify and validate and identify client

REQ-1.2: get all majors

REQ-1.3: get all colleges

REQ-1.4: get all job types

REQ-1.6: get all locations (state/regions)

REQ-1.7: get all OCECS regions (from regions table)

REQ-1.8: get quarters

REQ-1.9: get month/quarter mapping

3.2 Student Service Feature Set

3.2.1 Description

The primary role of the Student Service is to query the data contained in individual student records. This is a base service required for almost any additional service added to the system and is a top priority of the project.

3.2.2 Stimulus/Response Sequences

In order to make use of this service the consumer must be verified by the authentication module (REQ-1.1). If a user is not verified a failure message will be returned to the client. For all valid requests the responses will be returned in the SOAP protocol. Should an unexpected exception occur a SOAP Exception will be returned.

3.2.3 Functional Requirements

REQ-2.1: filter student records by a start and end quarters (inclusive start and end).

REQ-2.2: filter by job types

REQ-2.3: filter students by gender

REQ-2.4: filter by major(s) - can select multiple majors to query

REQ-2.5: filter by only NTID students

REQ-2.6: filter by only International students

- REQ-2.7: filter by students search status
- REQ-2.8: return report of only placed students
- REQ-2.9: return report of active, but not placed students
- REQ-2.10: return report of all students matching criteria
- REQ-2.11: search for single student by first name
- REQ-2.12: search for single student by last name
- REQ-2.13: search for single student by UID
- REQ-2.14: get a student's registration history
- REQ-2.15: provide all reports in simple XML format

3.3 Employer Service Feature Set

3.3.1 Description

The primary role of the Employer Service is to query the data associated with registered companies. This is a base service required for almost any additional service added to the system and is a top priority of the project.

3.3.2 Stimulus/Response Sequences

In order to make use of this service the consumer must be verified by the authentication module (REQ-1.1). If a user is not verified a failure message will be returned to the client. For all valid requests the responses will be returned in the SOAP protocol. Should an unexpected exception occur a SOAPException will be returned.

3.3.3 Functional Requirements

- REQ-3.1: get all companies
- REQ-3.2: get all account managers
- REQ-3.3: filter current employers with postings by start and end quarters
- REQ-3.4: filter by employers outside of the U.S
- REQ-3.5: filter employers with job postings by job types
- REQ-3.6: filter employers with job postings by location(state)
- REQ-3.7: filter employer by account manager
- REQ-3.8: filter employers with postings by major(s)
- REQ-3.9: filter employers with postings by college(s)
- REQ-3.10: return report of employers with postings (show info including contact)
- REQ-3.11: get postings of a employer
- REQ-3.12: get referrals for specific posting
- REQ-3.13: search for employer by name
- REQ-3.14: search for employer by account manager
- REQ-3.15: search for employer by contact
- REQ-3.15: filter employer by region
- REQ-3.16: filter employer by city/state
- REQ-3.17: filter employer by zip code
- REQ-3.18: provide all reports in simple XML format

3.4 Placement Service Feature Set

3.4.1 Description

The Placement Service is the first non-base service that relies on the base service of employer and student to retrieve and analyze the data and then return it to the user.

3.4.2 Stimulus/Response Sequences

In order to make use of this service the consumer must be verified by the authentication module (REQ-1.1). If a user is not verified a failure message will be returned to the client. For all valid requests the responses will be returned in the SOAP protocol. Should an unexpected exception occur a SOAPException will be returned.

3.4.3 Functional Requirements

- REQ-4.1: filter placements by start and end quarters
- REQ-4.2: filter by international placements
- REQ-4.3: filter placements with international students
- REQ-4.4: filter placements with international students working in home country
- REQ-4.5: filter placements with NTID students
- REQ-4.6: filter placements by location (state/region)
- REQ-4.7: filter placements by job type
- REQ-4.8: filter placements by major(s)
- REQ-4.9: filter placements by college(s)
- REQ-4.10: return report of placements from employer perspective
- REQ-4.11: return report of placements from student perspective(show SIS status)
- REQ-4.12: provide all reports in simple XML format

3.5 Activity Service Feature Set

3.5.1 Description

The Activity Service is another non-base service that relies on the base service of employer and student to retrieve and analyze the data and then return it to the user.

3.5.2 Stimulus/Response Sequences

In order to make use of this service the consumer must be verified by the authentication module (REQ-1.1). If a user is not verified a failure message will be returned to the client. For all valid requests the responses will be returned in the SOAP protocol. Should an unexpected exception occur a SOAPException will be returned.

3.5.3 Functional Requirements

- REQ-5.1: get all possible activity types
- REQ-5.2: filter activity records by start and end quarters
- REQ-5.3: filter activity records by single activity
- REQ-5.4: filter activity records by major(s)
- REQ-5.5: return report of student's activity by UID
- REQ-5.6: return report of activity records grouped by activity type
- REQ-5.7: return report of activity records grouped by college
- REQ-5.8: return report of activity records grouped by major
- REQ-5.9: return report of activity records grouped by quarter
- REQ-5.10: return report of activity records grouped by month
- REQ-5.11: return report of activity records grouped by office visit reason
- REQ-5.12: return report of activity records grouped by student name
- REQ-5.13: return report of activity records grouped by company
- REQ-5.14: return report of student job activities
- REQ-5.15: return report of all matching activity records

4. External Interface Requirements

4.1 User Interfaces

There will be no specific user interface for this system. The point of web services is that there can be multiple interfaces to the services as long as they conform to the SOAP standard and provide the necessary information as stipulated by the authentication contract. (see section 4.2)

4.2 Software Interfaces

The software interfaces for this product include the requests from the web services and the connection to the database. In order to make requests to most of the functions provided by the web services a client must be verified. This will be done using the current CAS authentication service, with the clients credentials being the ticket number from CAS. Using that number the web services will verify if a user has access to that function. Connections from outside clients will use the SOAP protocol.

In the event an exception occurs within one of the calls to the services, a subclass of the standard SOAP exception will be returned to the client. There will be two classes of SOAP exceptions. The first being exceptions associated with the verification of a user. These will be used to notify of a client that either the client needs to provide a CAS ticket or that the provided ticket is not valid. The second class of exceptions will be any other exceptions that occur in the process of responding to a request. These can include connection issues with the data-source, errors in reading inputs or other cases where the exceptions have not been handled. All exceptions of this second class will be logged and passed to the client to be handled as it sees fit.

The connection to the database will be done through a pooling module to increase speed and allow for an common interface for connection. The actual pooling of the database connections will be transparent to the web services and will appear as just a passthrough for the actual SQL calls.

4.3 Communications Interfaces

The primary communication interface to this product will be via HTTPS. Since the information being passed is very sensitive in nature all communication will be required to be encrypted. The communication over HTTPS will be done through the SOAP protocol.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

There are no specific performance requirements other than that the new system be faster than the existing system. Metrics will be kept to prove that similar queries in the new system will return results faster.

5.2 Security Requirements

Due to the sensitive nature of the information being provided by the system, all external calls must be made using SSL/HTTPS. Any calls that are not encrypted will be ignored.

5.3 Software Quality Attributes

The primary goal of this project is to create a system that is more maintainable and extensible than the previous system. This is being achieved by implementing a 3 tier/web-services architecture. Another benefit of the web-services is that the access points will be portable and available to any SOAP compatible client.

Finally, this system must be as accurate as the existing system. This implies that all like queries should result in the same dataset (assuming there are no flaws found in the existing system).

6. Future Development and Requirements

6.1 Explanation

This section describes possible future services or requirements of the system. These are not features that will be delivered at the completion of this project, but will be kept in mind when design and implementation decisions are made.

6.2 Future System Features

These are features that have been mentioned in meetings with the customer. This list is not extensive and may include features of future services to be added.

- “Login as” so administrator can view page as if they were logged in as a different user
- Mentor associations with students
- More “cluster” options (as opposed to just major/college clusters)
- Automated emailing of notices or other information
- Compensation options

6.3 Future Services

Future services that may be added to the system to allow for extended analysis or integration with other interfaces.

- Alert Service
- Portal Configuration Service